

# UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

Departamento de Ingeniería del Software e Inteligencia Artificial



## TESIS DOCTORAL

**Aplicación de algoritmos metaheurísticos en procesamiento de señales,  
imágenes y energías alternativas**

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

**Diego Alberto Oliva Navarro**

Directores

Gonzalo Pajares Martinsanz  
Erik Cuevas Jiménez

**Madrid, 2015**

**UNIVERSIDAD COMPLUTENSE DE MADRID**

**FACULTAD DE INFORMÁTICA**

**Departamento de Ingeniería del Software e Inteligencia Artificial**



**APLICACIÓN DE ALGORITMOS  
METAHEURÍSTICOS EN PROCESAMIENTO  
DE SEÑALES, IMÁGENES Y EN ENERGÍAS  
ALTERNATIVAS.**

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

**Diego Alberto Oliva Navarro**

Bajo la dirección de los doctores

Gonzalo Pajares Martinsanz

Erik Cuevas Jiménez

**Madrid, 2015**



UNIVERSIDAD COMPLUTENSE DE MADRID

Facultad de Informática



Departamento de Ingeniería del Software e Inteligencia Artificial

## **TESIS DOCTORAL**

# **APLICACIÓN DE ALGORITMOS METAHEURÍSTICOS EN PROCESAMIENTO DE SEÑALES, IMÁGENES Y EN ENERGÍAS ALTERNATIVAS**

Memoria para optar al grado de doctor presentada por:

Dn. Diego Alberto Oliva Navarro

Directores:

Dr. Gonzalo Pajares Martinsanz

Dr. Erik Cuevas Jiménez

Madrid (España), 2015





## **Agradecimientos**

La aventura del doctorado comenzó en mayo del 2011, cuando por primera vez pisaba suelo español, el objetivo principal era dar una plática sobre los temas que veníamos desarrollando en el laboratorio de la Universidad de Guadalajara. Después de conocer el grupo de trabajo del Dr. Gonzalo Pajares y de los consejos de los doctores Erik Cuevas, Daniel Zaldívar y Marco Pérez Cisneros, sentado en una habitación del hotel Princesa de Madrid comencé a visualizar como una mejor opción hacer el doctorado en la Universidad Complutense de Madrid. El proceso no ha sido sencillo, pero en cada una de las etapas que van desde obtener las becas, hasta el terminar de escribir este documento, pasando por los viajes y tramites, me he encontrado con el apoyo de las personas que aprecio.

Quiero agradecer primeramente a mis dos directores de tesis, ya que sin su apoyo y guía no hubiera sido posible concluir esta etapa de mi vida. En España al Dr. Gonzalo Pajares, por su gran amabilidad y apoyo. Desde el principio mostro gran interés en que realizara el doctorado bajo su tutela, siempre me brindo la información y herramientas necesarias para desarrollar mi investigación, en la parte personal siempre estuvo al tanto de mi bienestar, motivándome y guiándome para poder concluir satisfactoriamente la tesis doctoral. En México al Dr. Erik Cuevas a quien conozco ya desde hace varios años, por su continuo apoyo, motivación y enseñanza. Sin los consejos del Dr. Erik es muy probable que aún me encontrara perdido en el mundo científico. Por eso agradezco su disposición y la oportunidad que me dio de formar parte de su grupo de investigación. También por siempre darme todo lo necesario para poder desarrollar mi trabajo durante las estancias que realice en la Universidad de Guadalajara.

En la parte profesional a los doctores Marco Pérez Cisneros y Daniel Zaldívar, quienes durante años han estado al tanto de mi formación profesional. Además sus enseñanzas y consejos me han servido de guía a lo largo de estos años. A todos los compañeros del laboratorio de robótica y sistemas inteligentes de la Universidad de Guadalajara, gracias por todas las charlas que tuvimos ya que de ahí surgieron ideas bastante interesantes, pero sobre todo gracias por la motivación e interés. Al Dr. Valentín Osuna Enciso, a los maestros en ciencias Michel Dávalos, Felipe Sanción y Noé Ortega por las colaboraciones realizadas en conjunto y por la amistad que llevamos desde hace tiempo. En la Universidad Complutense de Madrid, a mis compañeros de doctorado Martín, José

Miguel, Juan y a la Dra. María Guijarro por las experiencias, aventuras y charlas que compartimos, es difícil olvidar esas excursiones a sembrar y tomar fotografías de los campos de cultivo.

Como no todo en la vida es trabajo, a lo largo de estos años tuve la oportunidad de compartir mi tiempo con muy buenos amigos es por eso que les agradezco su apoyo y motivación. A Marco Robles porque confiaste en mí aun sin conocerme y que además de compartir piso durante un tiempo nos volvimos muy buenos amigos, gracias por todos los consejos, las largas pláticas, los cafés interminables y por esos momentos en los que me escuchaste. Mary Calderón, por que llueve o truene has estado para motivarme y apoyarme en los momentos que más lo he necesitado. Es imposible nombrar a tantas personas con las que compartí buenos momentos, pero a todos ellos sin importar su nacionalidad, les doy las gracias por haber compartido su tiempo conmigo.

Mi familia ha sido una pieza clave en mi desarrollo profesional, sin el apoyo de mis padres José Luis y Ofelia nada de esto hubiera sido posible. Gracias por estar al pendiente de mí durante mis viajes, por preocuparse por mi salud, por animarme, por escucharme, por apoyarme en la medida de lo posible, por los consejos y ejemplos que aún me siguen dando. A mi hermano Daniel, con quien he compartido tantas aventuras, tantas cervezas y pláticas interminables. Siempre tengo presente el viaje que hicimos juntos y todo lo que hemos vivido. Si no hubiera sido por ustedes quizá no estaría el día de hoy escribiendo esto.

Hace casi dos años la vida puso en mi camino a la persona más especial que he conocido Gosia Kijak, gracias por formar parte de mi vida y de mi familia. Sin ti las cosas no serían tan divertidas como son, tu apoyo y amor incondicional me han dado la fortaleza y motivación necesaria para continuar con mis investigaciones. Has hecho todo por estar conmigo durante esta importante etapa de mi vida, gracias por soportarme, comprenderme, pero sobre todo gracias por amarme.

Antes de terminar quiero agradecer a dios por darme tantas bendiciones, por permitirme lograr hasta ahora cada uno de los objetivos que me he propuesto y por poner en mi camino a tantas personas que me han sabido apoyar.

Esta investigación ha sido financiada por el Gobierno de México a través del Consejo Nacional de Ciencia y Tecnología (CONACyT) con la beca doctoral 215517, la Secretaría de Educación Pública (SEP) por el programa de becas complemento 2013, 2014 y por el Instituto Jalisciense de la Juventud por el programa de becas impulso Jalisco 2011.

## Resumen

En años recientes se ha incrementado el interés por cómo la naturaleza da solución a diversos problemas como son: búsqueda de fuentes de alimento, organización de conjuntos de animales o insectos, atracción y repulsión de cargas, etc. De aquí surgen los algoritmos metaheurísticos (AM), los cuales simulan comportamientos naturales básicos para solucionar tareas complejas. El uso de estos métodos se ha extendido desde la optimización matemática hasta la ingeniería o la economía.

En este trabajo se abordan tres áreas de aplicación: 1) diseño de filtros digitales; 2) procesamiento de imágenes y 3) energías alternativas. Además, para algunas de estas aplicaciones se proponen modificaciones a los AM, con el fin de mejorar su rendimiento.

El primer AM que se describe es *Electromagnetism-like* (EMO), el cual está inspirado en los principios de atracción-repulsión de cargas eléctricas del electromagnetismo. Usando EMO se realizaron las siguientes aplicaciones: *a)* diseño de filtros IIR, donde los parámetros de los filtros se estiman mediante un proceso de optimización; *b)* detección de glóbulos blancos en imágenes médicas, aquí el proceso de identificación de leucocitos es analizado desde el punto de vista de la optimización, aplicando un detector de círculos basado en EMO que busca la forma circular que más se asemeje a las formas cuasi-circulares de las células sanguíneas; *c)* detección de objetos usando plantillas, aquí se busca la mayor similitud entre una plantilla seleccionada y una sección en la imagen donde esté contenida, para esto se emplea la correlación cruzada normalizada, cuya optimización se realiza mediante EMO; *d)* por último, se propone el uso de EMO para encontrar los mejores umbrales que permitan segmentar una imagen en función de su histograma, para esto se maximizan las ecuaciones de Otsu y Kapur. EMO es una poderosa alternativa de optimización, sin embargo, posee algunas deficiencias en la búsqueda local, lo cual en ciertas aplicaciones incrementa el costo computacional. Para esto, se propone dos modificaciones, una de ellas involucra el uso de la estrategia de aprendizaje conocida como *Opposition-Based Learning*. Mientras que la segunda es una modificación del proceso de búsqueda local, usando un muestreo alrededor de las mejores soluciones de cada iteración.

El segundo método que se estudió, es conocido como *Harmony Search* (HSA). El cual tiene como analogía, la improvisación que realizan los músicos para encontrar nuevas armonías. Este AM ha sido aplicado al problema de segmentación multinivel, donde el

objetivo es encontrar los mejores umbrales para binarizar imágenes que maximicen las funciones de Otsu y Kapur. Finalmente, estos valores son aplicados para determinar el mejor umbral en el histograma de la imagen, logrando con esto la segmentación de los píxeles originales como imagen binaria.

El tercer AM que se analizó es el *Artificial Bee Colony* (ABC), está inspirado en el comportamiento que tienen las abejas para encontrar nuevas fuentes de alimento. Este AM ha sido implementado para el diseño de celdas solares, para esto se emplea el modelo de diodo simple y diodo doble. El objetivo es encontrar la mejor configuración de parámetros que dé como resultados una buena relación corriente-voltaje.

Los métodos planteados se presentan como soluciones alternativas a los problemas selectos de las áreas de interés. Los resultados experimentales y comparaciones realizadas de forma independiente, comprueban su desempeño en términos de precisión y robustez. Del mismo modo, se comprueba que el uso de los AM es viable para una gran cantidad de aplicaciones orientadas al tratamiento de imágenes y de forma general en ingeniería.

**Palabras clave:** algoritmos metaheurísticos, filtros digitales, segmentación, imágenes médicas, modelado de celdas solares.

## Abstract

In recent years, the scientific community has increased the interest on the application of techniques inspired in the nature to solve some computational problems. Examples of such techniques are: search for food sources, animals or bugs organization, attraction-repulsion of charges. From these ideas the metaheuristic algorithms (MA) emerge to address the circumstances of mathematical optimization. They simulate different natural behaviors to solve complex tasks. The use of these methods has been extended from mathematical applications to implementations in different fields, including engineering.

In this work, three fields of application are explored and analyzed: 1) digital filter design; 2) image processing and 3) alternative energies. Moreover, some modifications are proposed for the MA in order to improve their performance.

The first MA implemented is Electromagnetism-like (EMO). This algorithm imitates the attraction–repulsion mechanism between charged particles in an electromagnetic field. Using EMO the following implementations are developed: *a)* IIR filter design, where filter parameters are estimated by an optimization process; *b)* white blood cells detection in medical images, where the identification of leukocytes is performed using a specific MA. A circle detector based on EMO is applied to obtain the best circular shape that matches with the quasi-circular shapes of blood cells; *c)* object detection using template matching, where EMO is used with the normal cross correlation as the objective function to find the best matching between a template and a section in the image; *d)* thresholding determination, where the use of EMO is proposed to find the best thresholds to segment the histogram of an image, to do this the Otsu's and Kapur's functions are maximized. EMO is a powerful optimization alternative, however, it has some shortcomings during the local search procedure in multidimensional applications where the computational cost is increased. For that reason, two modifications are proposed: one of them involves the use of a learning strategy, known as Opposition-Based Learning. Meanwhile, the second one is a modification of the local search using a sampling, at each iteration, around the best solutions found..

The second method studied is Harmony Search (HSA), that emulates the musicians improvisation looking for new harmonies while they are playing. HSA has been applied to multilevel segmentation problems, where the goal is to find the best threshold that

maximizes the Otsu's and Kapur's functions. Finally, such values are used for image segmentation based on histogram thresholding, obtaining a correct pixel classification.

The third MA analyzed is Artificial Bee Colony (ABC), which is inspired in the bee's behavior when they are looking for new food sources. This method has been implemented to solar cell engineering design using the single and double diode models. The goal is to find the best configuration of parameters, involved in the design, which provides a good current-voltage relationship.

The proposed methods are presented as alternative solutions for the selected problems in the specific areas. Experimental results and comparisons, carried out separately, prove their performance in terms of accuracy and robustness. On the same way, it is verified that the use of MA is feasible for different image processing and engineering applications.

**Keywords:** metaheuristic algorithms, digital filters, image segmentation, medical images, solar cells modelling.

## Índice general

<b>Capítulo 1</b>	<b>Introducción</b>	<b>1</b>
1.1	Antecedentes	1
1.2	Planteamiento del problema	11
1.3	Objetivos	15
1.3.1	Objetivos generales	16
1.3.2	Objetivos específicos	16
1.4	Metodología	16
1.5	Aportaciones de la investigación	17
1.6	Descripción de los capítulos	21
1.7	Resumen	23
<b>Capítulo 2</b>	<b>Marco teórico y estado del arte</b>	<b>25</b>
2.1	Visión general	25
2.2	Optimización y algoritmos metaheurísticos	26
2.2.1	Optimización	26
2.2.2	Métodos heurísticos y metaheurísticos	29
2.2.3	Algoritmos metaheurísticos (AM)	30
2.2.3.1	Historia	30
2.2.3.2	Conceptos en comunes	32
2.3	Fundamentos de <i>Opposition-based learning</i> (OBL)	32
2.3.1	Idea principal de OBL	33
2.3.2	Conceptos básicos de OBL	34
2.4	Filtrado digital de señales	34
2.4.1	Conceptos básicos de procesamiento digital de señales	34
2.4.2	Filtros FIR	36
2.4.3	Filtros IIR	37
2.4.4	Diseño de filtros digitales	37
2.5	Procesamiento de imágenes digitales	39
2.5.1	Sistemas de visión y procesamiento de imágenes	39
2.5.1.1	Procesamiento digital de imágenes	40
2.5.1.2	Relaciones básicas entre píxeles	40
2.5.1.3	Medidas de distancia	42
2.5.2	Histogramas	43



2.5.2.1	¿Qué es un histograma?	44
2.5.3	Segmentación	46
2.5.3.1	Segmentación por umbralización	48
2.5.3.2	Segmentación por crecimiento de regiones	50
2.5.4	Detección de líneas y curvas	51
2.5.4.1	Rasgos geométricos básicos	51
2.5.4.2	Estructuras en una imagen	52
2.5.4.3	La transformada de Hough	53
2.5.4.4	Transformada de Hough para la detección de círculos	60
2.5.5	Comparación y reconocimiento de imágenes	62
2.5.5.1	Comparación en imágenes a escala de grises	62
2.5.5.2	Distancia entre patrones	63
2.5.5.3	Distancia y correlación	64
2.5.5.4	Correlación cruzada normalizada	65
2.5.5.5	Coeficiente de correlación	66
2.6	Diseño y modelado de celdas solares	68
2.7	Resumen	69
<b>Capítulo 3 Algoritmo de optimización electromagnetism Like:</b>		<b>73</b>
<b>modificaciones y aplicaciones</b>		
3.1	Algoritmo de optimización <i>Electromagnetism-Like</i>	74
(EMO) 3.2	Algoritmo <i>Opposition-based Learning</i> (OBL)	80
3.2.1	Concepto de número opuesto	80
3.2.2	Concepto de un punto opuesto en el espacio	80
3.2.3	Optimización basada en lo opuesto	80
3.3	<i>Opposition-Based Electromagnetism-Like Optimization</i> (OBEMO)	82
3.3.1	Implementación del algoritmo OBEMO	83
3.3.2	Resultados experimentales	85
3.4	Modelado de filtros digitales	94
3.4.1	Filtros FIR e IIR	95
3.5	Modelado de filtros digitales usando el algoritmo EMO	95
3.5.1	Representación de las partículas	95
3.5.2	Función objetivo	96
3.5.3	Implementación	96

3.5.4	Resultados experimentales del algoritmo para identificación de parámetros y diseño de filtros IIR usando EMO	97
3.6	Detección de círculos en imágenes digitales usando el algoritmo EMO	112
3.6.1	Pre-procesamiento de los datos	112
3.6.2	Representación de las partículas	113
3.6.3	Función objetivo	114
3.6.4	Detección de círculos usando el algoritmo EMO	116
3.7	Detector de glóbulos blancos usando el algoritmo EMO	118
3.7.1	Pre-procesamiento de la imagen	118
3.7.2	El detector de círculos basado en EMO modificado	119
3.7.3	Un ejemplo numérico	122
3.7.4	Resultados experimentales de la detección de glóbulos blancos usando EMO	124
3.7.4.1	Comparaciones con otros métodos	125
3.8	Localización y detección de objetos por medio de plantillas	132
3.9	Modificaciones del algoritmo EMO para dar solución al problema de localización y detección de objetos mediante el uso de plantillas	135
3.9.1	El nuevo proceso de búsqueda local para EMO	136
3.9.2	Incorporación de la memoria	142
3.10	Localización y detección de objetos mediante el uso de plantillas empleando EMO como estrategia de búsqueda	142
3.10.1	Implementación	143
3.10.2	Resultados experimentales	145
3.11	Resumen	151
<b>Capítulo 4</b>	<b>Algoritmo de optimización <i>Harmony Search</i>: segmentación multinivel de imágenes digitales</b>	<b>153</b>
4.1	Algoritmo de optimización <i>Harmony Search</i> (HSA)	154
4.2	Segmentación multinivel de imágenes digitales usando <i>Harmony Search</i> (MTHSA)	157
4.2.1	Representación de las armonías	158
4.2.2	Criterio de la entropía (método de Kapur)	160
4.3	Segmentación multinivel de imágenes digitales usando <i>Harmony Search</i> (MTHSA)	162
4.3.1	Representación de las armonías	162

4.3.2	Implementación del algoritmo MTHSA	162
4.3.3	Configuración de los parámetros de MTHSA	163
4.3.4	Resultados experimentales para la segmentación multinivel usando HSA	165
4.3.4.1	Resultados de HSA usando la ecuación de Otsu	167
4.3.4.2	Resultados de HSA usando la ecuación de Kapur	174
4.3.4.3	Comparaciones	174
4.3	Resumen	184
<b>Capítulo 5</b>	<b>Algoritmo de optimización <i>Artificial Bee Colony</i>: diseño de celdas solares</b>	<b>185</b>
5.1	Algoritmo de optimización <i>Artificial Bee Colony</i> (ABC)	186
5.1.1	Descripción del algoritmo ABC	186
5.1.2	Inicialización de la población	186
5.1.3	Enviar abejas obreras	187
5.1.4	Selección de las fuentes de alimento por medio de las abejas espectadoras	188
5.1.5	Determinar las abejas exploradoras	188
5.1.6	Algoritmo ABC	189
5.2	Diseño de celdas solares	189
5.2.1	Modelo de doble diodo	189
5.2.2	Modelo de diodo simple	191
5.2.3	Identificación de parámetros de celdas solares, como un problema de optimización	192
5.3	Identificación de parámetros de celdas solares usando el algoritmo ABC	193
5.3.1	Planteamiento del problema	194
5.3.2	Enfoque computacional para el diseño de celdas solares	194
5.4	Resultados experimentales	195
5.4.1	Resultados experimentales del algoritmo ABC	196
5.4.2	Comparaciones con otros enfoques similares	199
5.5	Resumen	202
<b>Capítulo 6</b>	<b>Conclusiones y trabajo futuro</b>	<b>205</b>
6.1	Conclusiones específicas	205
6.1.1	Conclusiones: <i>Opposition-Based Electromagnetism-Like</i>	206

	<i>Optimization (OBEMO)</i>	
6.1.2	Conclusiones: Modelado de filtros digitales usando el algoritmo EMO	206
6.1.3	Conclusiones: Detector de glóbulos blancos usando el algoritmo EMO	207
6.1.4	Conclusiones: Localización y detección de objetos mediante el uso de plantillas empleando EMO como estrategia de búsqueda	208
6.1.5	Conclusiones: Segmentación multinivel de imágenes digitales usando <i>Harmony Search</i>	209
6.1.6	Conclusiones: Identificación de parámetros de celdas solares usando el algoritmo ABC	209
6.2	Conclusiones generales	210
6.3	Trabajo futuro	211
	<b>English summary</b>	<b>213</b>
	<b>Bibliografía</b>	<b>243</b>



## Índice de abreviaturas

A/D	Analógico-Digital
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
AE	Algoritmo Evolutivos
AIA	Artificial Immune Algorithm
AM	Algoritmos Metaheurísticos
ANCC	Promedio de NCC
ANOVA	Análisis de la varianza
ASL	Promedio del número de localidades de búsqueda
BFA	Bacterial Foraging Algorithm
BL	Búsqueda Local
BM	Técnica de ajuste de bloques para estimación de movimiento
BSVs	Boundary Support Vectors
BW	Ancho de banda
CCD	Dispositivo de carga acoplada
CF	Celdas Fotovoltaicas
CMOS	Semiconductor complementario de óxido metálico
CS	Celdas Solares
CT	Tiempo computacional
D/A	Digital-Analógico
DD	Diodo Doble
DE	Differential Evolution
DEM	Diffused Expectation-Maximization
DR	Índice de detección de leucocitos
DS	Diodo Simple
EMO	Electromagnetism-Like Optimization
EMO-BL	EMO con búsqueda local modificada
EMO-O	EMO Original
ES	Energía Solar
FAR	Índice de falsa alarma para detección de leucocitos
FCNN	Fuzzy Cellular Neural Network
FD	Filtros Digitales

FEMO	Fixed pattern search EMO
FIR	Finite Impulse Response
FM	Memoria de afinidad
GA	Genetic Algorithms
GAB	Based on Genetic Algorithms
GB	Glóbulos Blancos
GSA	Gravitational Search Algorithm
HEMO	Hybridizing EMO
HM	Harmony Memory (memoria de armonías)
HMCR	Parámetro de consideración de la memoria de armonía
HSA	Harmony Search Algorithm
HSI	Espacio de color (Hue-Saturation-Intensity)
HT	Transformada de Hough
ICA	Imperialist Competitive Algorithm
IIR	Infinite Impulse Response
IO	Iterative Otsu
I-V	Relación corriente-voltaje
JCR	Journal Citation Reports
JPEG	Formato de imágenes (Joint Photographic Experts Group)
LA	Learning Automata
LTI	Lineales invariantes en el tiempo
MAE	Error absoluto medio
MBE	Error de sesgo medio
MCA	Mid-point Circle Algorithm
MSE	Mean Square Error
MTHSA	Segmentación multinivel usando HSA
NCC	Correlación cruzada normalizada
NI	Número de iteraciones
Nim	Número de improvisaciones
NMAE	Error absoluto medio normalizado
NMBE	Error de sesgo medio normalizado
NRMSE	Error cuadrático medio normalizado
OBEMO	Opposition Based Electromagnetism-Like
OBL	Opposition Based Learning

OG	Optimización Global
PAR	Proporción de ajuste de tono
PDI	Procesamiento Digital de Imágenes
PDS	Procesamiento Digital de Señales
PRMSE	Promedio del error cuadrático medio
PSNR	Peak to signal ratio
PSO	Particle Swarm Optimization
R.T.C.	Radio Technique Compelec, Francia
RGB	Espacio de color (Red, Green and Blue)
RMSE	Error cuadrático medio
SA	Simulated Annealing
SAD	Suma de diferencias absolutas
SC	Soluciones Candidatas
SR	Rango de éxito en la detección de objetos
SSD	Suma de diferencias cuadráticas
STD	Desviación estándar
SVM	Support Vector Machines
TH	Thresholding
th	Threshold
TM	Template Matching
TS	Tabu Search
TSP	Traveling Salesman Problem
VC	Visión por Computador





## Índice de figuras

Figura 2.1	Espacio de búsqueda factible	27
Figura 2.2	Clasificación de los problemas de optimización	28
Figura 2.3	Clasificación de los algoritmos de optimización	29
Figura 2.4	Diagrama de flujo para procesar una señal analógica	36
Figura 2.5	Taxonomía de los principales métodos para el diseño de filtros digitales	38
Figura 2.6	Elementos que componen el sistema de procesamiento de imagen	39
Figura 2.7	Definición de la vecindad de un píxel P, (a) Vecindad 4 y (b) vecindad 8	41
Figura 2.8	Influencia de la vecindad en la conectividad de píxeles: (a) imagen considerando conectividad-4, (b) imagen considerando conectividad-8	42
Figura 2.9	Tipos de distancia para medir las relaciones posicionales entre píxeles: (a) definición de los píxeles; (b) distancia euclidiana; (c) distancia <i>city-block</i> ; (d) distancia <i>chessboard</i>	43
Figura 2.10	Imagen de intensidad en escala de grises de 8 bits con su respectivo histograma	45
Figura 2.11	Ejemplo de representación de un histograma	45
Figura 2.12	Tres diferentes imágenes que producen el mismo histograma	46
Figura 2.13	Ejemplo de aplicación de la segmentación bi-nivel sobre una imagen, considerando $th=80$ , $p_0=0$ y $p_1=1$ ; (a) imagen en escala de grises; (b) imagen binarizada; (c) y (d) sus respectivos histogramas	47
Figura 2.14	(a), (b), (c), (d), resultados tras aplicar un algoritmo de segmentación por contornos activos. (e) y (f) son los resultados tras segmentar por superpíxeles	51
Figura 2.15	Imágenes de ejemplo en las cuales el ojo humano puede distinguir los objetos existentes a pesar de los puntos añadidos como ruido a la imagen	53
Figura 2.16	Los objetos paramétricos aparecen como líneas, círculos o elipses, aparecen con frecuencia en las imágenes digitales	54
Figura 2.17	Dos puntos $p_1$ y $p_2$ que pertenecen a la misma línea, por lo que para determinar la ecuación de la recta habrá que estimar $k$ y $d$	55
Figura 2.18	Conjunto de rectas que pasan a través de un punto $p_0$ en común. Todas las posibles líneas $L_p$ que pasan por $p_0$ tienen la expresión $y_0 = kx_0 + d$	56
Figura 2.19	Relación entre el espacio de parámetros de la imagen y el espacio de	57

	parámetros de Hough; (a) espacio de parámetros de la imagen; (b) espacio de parámetros de Hough	
Figura 2.20	Idea fundamental de la transformada de Hough; (a) espacio de parámetros de la imagen y (b) acumulador del espacio de parámetros de Hough. La matriz de registros de acumulación es la versión discreta del espacio de Hough	58
Figura 2.21	Diferentes tipos de parámetros para caracterizar líneas; (a) se muestra la singularidad para el caso de una línea vertical (ecuación (2.20)); (b) presenta el uso del modelo que evita el error computacional (ecuación (2.25))	58
Figura 2.22	(a) Espacio de parámetros de la imagen y (b) espacio de parámetros de Hough usando el modelo de la recta de la ecuación (2.26)	60
Figura 2.23	Parámetros que definen a un círculo	61
Figura 2.24	Resultados parciales tras aplicar la transformada de Hough para detección de círculos: (a) imagen original; (b) bordes de (a); (c) representación tridimensional del arreglo de registros acumuladores; (d) detección de los máximos registrados	62
Figura 2.25	Geometría del problema de comparación de imágenes	63
Figura 2.26	Medición de la distancia $d$ entre funciones de dos dimensiones. La imagen de referencia está posicionada sobre el píxel $I(r,s)$	64
Figura 2.27	Modelos de celdas solares: (a) modelo de doble diodo y (b) modelo de diodo simple	68
Figura 3.1	Principio de superposición	77
Figura 3.2	Ley de Coulomb	78
Figura 3.3	Procedimiento de la optimización basada en lo opuesto: (a) función a optimizar, (b) contorno de la función a optimizar	81
Figura 3.4	Diagrama de flujo para: (a) algoritmo EMO y (b) OBEMO	83
Figura 3.5	Comparación del proceso del optimización para dos de las funciones de prueba originales: (a) $f_3$ y (b) $f_6$	89
Figura 3.6	Comparación del proceso del optimización para dos de las funciones de prueba multimodales: (a) $f_{12}$ y (b) $f_{14}$	93
Figura 3.7	Proceso de identificación de los parámetros de un filtro digital usando algoritmos metaheurísticos	94

Figura 3.8	(a) evolución del error cuadrático medio y (b) parámetros $a$ y $b$ a lo largo de las iteraciones, en ambos casos para la planta de segundo orden con un modelo de primer orden	99
Figura 3.9	(a) evolución del error cuadrático medio y (b) parámetros $a_1$ , $a_2$ y $b$ a lo largo de las iteraciones, en ambos casos para la planta de segundo orden con un modelo de segundo orden	101
Figura 3.10	(a) evolución del error cuadrático medio y (b) parámetros $a_1$ , $a_2$ y $b$ a lo largo de las iteraciones, en ambos casos para la planta de tercer orden con un modelo de segundo orden	103
Figura 3.11	(a) evolución del error cuadrático medio y (b) parámetros $a_1, a_2, a_3, a_4$ y (c) parámetros $b_0, b_1, b_2, b_3, b_4$ a lo largo de las iteraciones, para un filtro de orden 4	105
Figura 3.12	(a) comparación de los valores de MSE de los algoritmos EMO y PSO y (b) comparación de los parámetros $a$ y $b$ encontrados por EMO y PSO durante 30000 iteraciones. En ambos casos para una planta de segundo orden modelada con un filtro de primer orden	107
Figura 3.13	(a) comparación de los valores de MSE de los algoritmos EMO y PSO y (b) comparación de los parámetros $a$ y $b$ encontrados por EMO y PSO durante 30000 iteraciones. En ambos casos para una planta de segundo orden modelada con un filtro de segundo orden	109
Figura 3.14	Graficas de los parámetros del denominador del filtro de orden 4 para los algoritmos (a) EMO y (b) PSO	110
Figura 3.15	Graficas de los parámetros del numerador del filtro de orden 4 para los algoritmos (a) EMO y (b) PSO	111
Figura 3.16	Comparación de los valores de MSE de los algoritmos EMO y PSO	112
Figura 3.17	Círculo candidato (partícula cargada) construido a partir de la combinación de los puntos de borde $p_{e_1}$ , $p_{e_2}$ y $p_{e_3}$	114
Figura 3.18	Procedimiento para evaluar la función objetivo. (a) es la imagen original, (b) muestra la forma virtual $\mathbf{H}$ que corresponde a $\mathbf{C}$ y (c) tiene las coincidencias entre ambas imágenes	116
Figura 3.19	Analogía con la ley de Coulomb	117
Figura 3.20	Pre-procesamiento: (a) imagen original de frotis sanguíneo, (b) imagen	120

	segmentada obtenida por DEM y (c) el mapa de bordes obtenido por medio del de detector de bordes morfológico	
Figura 3.21	Proceso de detección de glóbulos blancos. (a) imagen de frotis sanguíneo, (b) imagen segmentada, (c) mapa de bordes, (d) círculo detectado usando la función objetivo original, los puntos rojos muestran la coincidencia entre el círculo candidato y el mapa de bordes. (e) círculo detectado usando la nueva función objetivo, los puntos amarillos representan los bordes sin coincidencia. (f) resultado final	122
Figura 3.22	Ejemplo numérico de detección de una única célula sanguínea: (a) imagen usada como ejemplo, (b) imagen segmentada, (c) mapa de bordes, (d) partículas iniciales, (e) fuerzas ejercidas sobre $C_{3,0}$ , (f) nueva posición de $C_{3,0}$ , (g) posición de las partículas después de la primer generación, (h) configuración final de las partículas después de 20 generaciones, (i) resultado final sobre la imagen original	124
Figura 3.23	Imágenes resultantes de la primera prueba aplicando el detector de glóbulos sanguíneos basado en EMO: (a) imagen original, (b) imagen segmentada por el algoritmo DEM, (c) mapa de bordes y (d) leucocitos detectados	125
Figura 3.24	Imágenes resultantes de la primera prueba aplicando el detector de glóbulos sanguíneos basado en EMO: (a) imagen original, (b) imagen segmentada por el algoritmo DEM, (c) mapa de bordes y (d) leucocitos detectados	127
Figura 3.25	Ejemplos de imágenes incluidas en el conjunto de prueba para la comparación de robustez. (a) Imagen contaminada con el 10% de ruido sal y pimienta y (b) imagen con $\sigma=10$ de ruido Gaussiano	128
Figura 3.26	Comparación de los métodos de Wang y EMO para detección de células sanguíneas en imágenes médicas. (a) Imagen original, (b) resultados de la detección usando en el algoritmo de Wang, (c) detección tras aplicar el algoritmo EMO	131
Figura 3.27	Resultados de comparación para la detección de glóbulos blancos mostrando: (a) el algoritmo de Wang después de 400 ciclos y (b) el detector basado en EMO considerando 1000 ciclos	132
Figura 3.28	Geometría del problema de localización y reconocimiento de objetos	133

	mediante plantillas	
Figura 3.29	Proceso de localización y detección de objetos en imágenes mediante el uso de plantillas, (a) imagen fuente, (b) imagen plantilla, (c) valores de NCC codificados en color y (c) superficie multimodal de NCC	135
Figura 3.30	Funcionamiento del nuevo mecanismo de búsqueda local, (a) opera considerando la población inicial modificada $Y_1$ , (b) considerando población modificada $Y_{15}$ después de 15 iteraciones y (c) opera considerando la población modificada $Y_{25}$ después de 25 iteraciones	141
Figura 3.31	Patrón de búsqueda generado por el algoritmo EMO, los puntos verdes representan las posiciones evaluadas, los puntos azules indican que los puntos ya han sido visitados y están almacenadas en la <b>FM</b> . El punto rojo es el que tiene la similitud óptima	145
Figura 3.32	Comparación de los valores de NCC para las imágenes: (a) perro, (b) soccer, (c) Waldo, (d) aeropuerto, (e) ciudad y (f) PCB, usando los cinco diferentes algoritmos	150
Figura 4.1	(a) <i>Cameraman</i> , (c) <i>Lena</i> , (e) <i>Baboon</i> , (g) <i>Hunter</i> y (i) <i>Butterfly</i> , imágenes seleccionadas. (b), (d), (f), (h) y (j) histogramas correspondientes a cada imagen	166
Figura 5.1	Modelo de doble diodo para las celdas solares	190
Figura 5.2	Modelo de diodo simple para las celdas solares	191
Figura 5.3	Para el modelo de diodo doble: (a) voltaje medido frente a corriente calculada por ABC para diferentes valores de temperatura, (b) voltaje medido frente a potencia obtenida por ABC a diferentes temperaturas, (c) evolución del RMSE. Para el modelo diodo simple: (d) voltaje medido frente a corriente calculada por ABC para diferentes valores de temperatura, (e) voltaje medido frente a potencia obtenida por ABC a diferentes temperaturas, (f) evolución del RMSE	200
Figura 5.4	Comparación de los métodos ABC, HS, PSO, GA y BFA para el modelo de doble diodo en (a) voltaje medido frente a la corriente calculada y (b) voltaje medido frente a la potencia y para el modelo de diodo simple en (c) voltaje medido frente a la corriente calculada, (d) voltaje medido frente a la potencia	202



## Índice de tablas

Tabla 2.1	Taxonomía de los algoritmos metaheurísticos	31
Tabla 2.2	Características de las señales	35
Tabla 2.3	Métodos de segmentación por búsqueda de umbrales	49
Tabla 2.4	Relación existente de puntos y líneas entre los diferente espacios de parámetros	56
Tabla 2.5	Algunos métodos para el modelado y diseño de celdas solares	70
Tabla 3.1	Funciones de optimización de prueba, correspondientes al conjunto de original	86
Tabla 3.2	Conjunto de funciones de prueba multidimensionales	87
Tabla 3.3	Resultados comparativos para los algoritmos EMO, OBEMO, HEMO y FEMO considerando las funciones originales (Tabla 3.1)	88
Tabla 3.4	Resultados de la prueba rangos de Wilcoxon considerando el “promedio del mejor valor de $f(\mathbf{x})$ ”	90
Tabla 3.5	Resultados de la prueba rangos de Wilcoxon considerando el “promedio de iteraciones <i>MAXITER</i> ”	91
Tabla 3.6	Resultados comparativos para los algoritmos EMO, OBEMO, HEMO y FEMO considerando las funciones multimodales (Tabla 3.2)	92
Tabla 3.7	Resultados de la prueba rangos de Wilcoxon considerando el “promedio del mejor valor de $f(\mathbf{x})$ ” de las funciones multimodales	93
Tabla 3.8	Resultados de la prueba rangos de Wilcoxon considerando el “promedio de iteraciones <i>MAXITER</i> ” de las funciones multimodales	94
Tabla 3.9	Valores encontrados por el algoritmo EMO y valores reales del modelo (primer problema)	99
Tabla 3.10	Valores encontrados por el algoritmo EMO y valores reales del modelo (segundo problema)	101
Tabla 3.11	Valores encontrados por el algoritmo para los parámetros del numerado y los del denominador	106
Tabla 3.12	Valores comparativos de los parámetros $a$ , $b$ , y $J$ encontrados tanto por el algoritmo EMO como el PSO	107
Tabla 3.13	Valores comparativos de los parámetros $a$ , $b$ , y $J$ encontrados tanto por el algoritmo EMO como el PSO	108



Tabla 3.14	Valores encontrados por el algoritmo EMO y PSO para los parámetros del numerador y los del denominador de un filtro de orden 4	111
Tabla 3.15	Parámetros del algoritmo EMO usados para la detección de leucocitos en imágenes medicas	122
Tabla 3.16	Comparación del rendimiento en la detección de leucocitos para los enfoques BSV, IO, Wang, BGA y el método propuesto basado en EMO	126
Tabla 3.17	Comparación en la detección de leucocitos entre los enfoques BSV, IO, Wang, BGA y el método propuesto basado en EMO, considerando un conjunto de 30 imágenes contaminadas con diferentes niveles de ruido sal y pimienta	129
Tabla 3.18	Comparación en la detección de leucocitos entre los enfoques BSV, IO, Wang, BGA y el método propuesto basado en EMO, considerando un conjunto de 30 imágenes contaminadas con diferentes niveles de Gaussiano	129
Tabla 3.19	Error en la estimación del tamaño de las células después aplicar el algoritmo EMO y el método de Wang para detectar un leucocito contenido dentro de una imagen de frotis sanguíneo. El error es promediado a partir de 20 experimentos	132
Tabla 3.20	Algoritmo EMO para localización y detección de objetos mediante plantillas, aplicado a diferentes tipos de imágenes	146
Tabla 3.21	Comparación del rendimiento de los método ICA, PSO, GA EMO y EMO con BL modificada para las imágenes experimentales presentadas en la Tabla 3.20	149
Tabla 3.22	Valores $p$ , producidos por la prueba de Wilcoxon comparando los valores ASL de la siguiente manera: EMO-BL contra ICA, EMO-BL contra PSO, EMO-BL contra GA y EMO-BL contra EMO-O	151
Tabla 4.1	Niveles de los factores empleados para el método de diseño factorial	164
Tabla 4.2	Parámetros de HS obtenidos por el método de diseño factorial	165
Tabla 4.3	Resultados de aplicar el algoritmo HS usando la función de Otsu al conjunto de imágenes de prueba	168
Tabla 4.4	Resultados de aplicar el algoritmo HS usando la función de Otsu a la imagen de prueba: <i>Cameraman</i>	169
Tabla 4.5	Resultados de aplicar el algoritmo HS usando la función de Otsu a la imagen de prueba: <i>Lena</i>	170
Tabla 4.6	Resultados de aplicar el algoritmo HS usando la función de Otsu a la	171

	imagen de prueba: <i>Baboon</i>	
Tabla 4.7	Resultados de aplicar el algoritmo HS usando la función de Otsu a la imagen de prueba: <i>Hunter</i>	172
Tabla 4.8	Resultados de aplicar el algoritmo HS usando la función de Otsu a la imagen de prueba: <i>Butterfly</i>	173
Tabla 4.9	Resultados de aplicar el algoritmo HS usando la función de Kapur al conjunto de imágenes de prueba	175
Tabla 4.10	Resultados de aplicar el algoritmo HS usando la función de Kapur a la imagen de prueba: <i>Cameraman</i>	176
Tabla 4.11	Resultados de aplicar el algoritmo HS usando la función de Kapur a la imagen de prueba: <i>Lena</i>	177
Tabla 4.12	Resultados de aplicar el algoritmo HS usando la función de Kapur a la imagen de prueba: <i>Baboon</i>	178
Tabla 4.13	Resultados de aplicar el algoritmo HS usando la función de Kapur a la imagen de prueba: <i>Hunter</i>	179
Tabla 4.14	Resultados de aplicar el algoritmo HS usando la función de Kapur a la imagen de prueba: <i>Butterfly</i>	180
Tabla 4.15	Valores <i>P</i> obtenidos por la prueba de Wilcoxon, comparando los resultados de PSNR de Otsu contra los de Kapur	182
Tabla 4.16	Comparaciones entre los algoritmos MTHS, GA, PSO y BFA aplicados sobre las imágenes de prueba seleccionadas usando el método de Otsu	183
Tabla 4.17	Comparaciones entre los algoritmos MTHS, GA, PSO y BFA aplicados sobre las imágenes de prueba seleccionadas usando el método de Kapur	183
Tabla 5.1	Rangos de los parámetros de las celdas solares	192
Tabla 5.2	Parámetros de configuración del algoritmo ABC	196
Tabla 5.3	Parámetros obtenidos por el método basado en ABC para los modelos de doble diodo y de diodo simple	197
Tabla 5.4	Mediciones en la terminal de la celda solar, valores estimados por el ABC, error relativo y error relativo normalizado para los modelos de celda solar de doble diodo y diodo simple	198
Tabla 5.5	Índices de rendimiento para los modelos de celdas solares: doble diodo, y diodo simple	199
Tabla 5.6	Comparación de los resultados para el modelo de doble diodo	201
Tabla 5.7	Comparación de los resultados para el modelo de diodo simple	201



## Índice de algoritmos

Algoritmo 3.1	Inicialización	75
Algoritmo 3.2	Búsqueda local	76
Algoritmo 3.3	Calculo del vector de fuerza total	78
Algoritmo 3.4	Movimiento	79
Algoritmo 3.5	Nueva Búsqueda local	139
Algoritmo 4.1	Harmony Search básico	157
Algoritmo 4.2	Harmony Search para múltiples umbrales (MTHSA)	163
Algoritmo 5.1	ABC	189
Algoritmo 5.2	Diseño de celdas solares usando ABC	195



# Capítulo 1

## Introducción

---

### 1.1 Antecedentes

La Optimización Global (OG) [1,2] tiene aplicaciones en diversas áreas de ciencia [3], economía [4,5], ingeniería [6] y otras donde se requiere por definición el uso de modelos matemáticos [7,8]. Precisamente, el presente trabajo de investigación tiene como objetivo el estudio de métodos de OG muy apropiados para su utilización eficiente en tres áreas específicas de aplicación, a saber:

- Diseño de filtros digitales
- Procesamiento de imágenes
- Energías alternativas

Simultáneamente, en el marco de la investigación, se plantea la mejora de la eficiencia de uno de los algoritmos metaheurísticos de interés.

De forma general, el objetivo de OG es encontrar el óptimo global de una función objetivo. La cual está definida sobre un espacio de búsqueda acotado. Las dificultades asociadas con el uso de los métodos matemáticos tradicionales, sobre problemas de OG, han contribuido al desarrollo de soluciones alternativas. Por ejemplo, las técnicas de programación lineal y dinámica usualmente fallan o alcanzan óptimos locales, al resolver problemas de tiempo polinomial no determinista complejos (NP-hard). Los cuales tienen como característica un gran número de variables y funciones objetivo no lineales. Con el fin de superar tales problemas, los investigadores han propuesto la búsqueda de soluciones óptimas usando algoritmos metaheurísticos (AM).

Los AM son métodos estocásticos, que simulan la metáfora de fenómenos biológicos o físicos. El núcleo de tales métodos, radica en el análisis del comportamiento colectivo de agentes relativamente simples, que trabajan sobre sistemas descentralizados. Tales sistemas, usualmente reúnen una población de agentes, que pueden comunicarse entre sí mientras comparten un ambiente común. A pesar de que un algoritmo de control no centralizado, regula el comportamiento del agente, dicho agente puede resolver tareas complejas, analizando un modelo global dado y reuniendo los esfuerzos de cooperación de otros elementos. Por lo tanto, un nuevo comportamiento global evoluciona a partir de la interacción entre los agentes. Esto puede observarse en ejemplos típicos, que incluyen: colonias de hormigas, manadas de animales, bandadas de aves, bancos de peces, abejas, bacterias, partículas cargadas y muchos más. Algunos otros algoritmos de optimización metaheurísticos han sido propuestos recientemente, entre ellos destacan:

- *Genetic Algorithms* (GA) [9], los cuales emplean operadores que simulan la genética para modificar una población de individuos, hasta encontrar la mejor solución.
- *Particle Swarm Optimization* (PSO) [10], en el cual se crea una población de partículas que es modificada con operadores basados en el comportamiento de las bandadas de aves o los bancos de peces.
- *Differential Evolution* (DE) [11], usa diversos operadores de mutación y cruce para crear nuevas soluciones. Las cuales compiten con una población previamente creada hasta encontrar la mejor solución.
- *Ant Colony Optimization* (ACO) [12], está basado en las feromonas que las hormigas emplean al desplazarse en busca de las mejores opciones de alimentación.

Dentro de las propuestas de AM, el algoritmo de optimización *Electromagnetism-like* (EMO) es un método basado en población relativamente nuevo. El cual fue propuesto por vez primera por Birbil y Fang [13] para resolver modelos de optimización continuos usando variables acotadas. El algoritmo imita el mecanismo de atracción y repulsión entre partículas cargadas, tal como sucede en un campo electromagnético. Cada partícula representa una solución y lleva consigo una cierta cantidad de carga, la cual es proporcional a la calidad de la solución (función objetivo). A su vez, las soluciones son definidas por vectores de posición, los cuales asignan posiciones reales para las partículas dentro de un espacio multidimensional. Además, los valores de la función objetivo de cada partícula, son calculados considerando tales vectores de posición. Cada partícula ejerce fuerzas de atracción o repulsión sobre los otros miembros de la población. La fuerza resultante que actúa sobre un miembro de la población, se usa para actualizar su siguiente posición. Claramente, la idea subyacente detrás de la metodología EMO, es mover las partículas hacia una solución óptima por medio de las fuerzas de atracción o repulsión. A diferencia de otras técnicas metaheurísticas como GA, PSO, DE y ACO cuyos miembros de la población intercambian materiales o información entre ellos. EMO asume que cada partícula es influenciada por el resto de los miembros de la población, imitando métodos como PSO o ACO. Aunque EMO comparte características con PSO y ACO, existen trabajos recientes que exhiben sus capacidades de convergencia, manteniendo a su vez buena precisión para encontrar los valores óptimos [14-18]. Cabe mencionar que EMO ha sido exitosamente aplicado para dar solución a diversos problemas de ingeniería. Destacan entre otros los siguientes: organización del manejo de materiales en las líneas de producción (*flow shop scheduling*) [19], comunicaciones [20], planeación de rutas de vehículos [21], diseño de circuitos electrónicos [22], entrenamiento de redes neuronales [23] y procesamiento digital de imágenes [24].

De forma similar, un interesante método de optimización es: *Harmony Search Algorithm* (HSA), introducido por Geem et al. [25]. HSA es un algoritmo metaheurístico basado en la metáfora del proceso de improvisación, que ocurre cuando los músicos buscan mejores estados de armonía. Dentro del funcionamiento del HSA, se genera una nueva solución candidata, teniendo en cuenta todas las soluciones existentes. El vector solución es análogo a la armonía musical. Mientras que los esquemas de búsqueda local y global, son una analogía de la improvisación de los músicos. En comparación con AM tradicionales, HSA impone pocos requisitos matemáticos. Por lo que puede ser fácilmente adaptado, para resolver diferentes tipos de retos de optimización en ingeniería [26]. Además, se han



realizado diversas comparaciones numéricas. Las cuales demuestran que la convergencia de HSA es más rápida que la de GA [27], por lo cual resulta altamente atractivo. Este algoritmo, también ha sido exitosamente aplicado para la solución de una gran cantidad de problemas de optimización prácticos, destacando: la optimización discreta y continua de estructuras [28], la estimación de parámetros del modelo no lineal de Muskingum [29], el diseño óptimo de redes de distribución de agua [30], la planeación de rutas de vehículos [31], el suministro económico de la combinación de calor y potencia [32], el diseño de cuadros de metal [33] y el procesamiento de imágenes [34]. Aunque el HSA estándar presenta importantes características de búsqueda, una gran cantidad de modificaciones han sido propuestas en la literatura, con la intención de mejorar las capacidades del propio algoritmo [35].

Otro interesante método de optimización dentro de los AM es *Artificial Bee Colony* (ABC) [36]. El cual se inspira en el comportamiento inteligente de las abejas mieleras (se usará la palabra abeja para referirnos a este tipo de insecto). ABC está formado por tres componentes esenciales, las fuentes de alimento, la cantidad de néctar, y las distintas clases de abejas. La posición de cada fuente de alimento, representa una posible solución para el problema que se está tratando. La cantidad de néctar de una fuente de alimento, se representa con el valor de una función de afinidad e indica la calidad de dicha solución. Cada tipo de abeja simboliza una operación, en particular para generar nuevas posiciones de fuentes de alimento. El algoritmo ABC se inicia generando aleatoriamente una población inicial de fuentes de alimento. Después de la inicialización, la población es evaluada en la función objetivo obteniéndose la cantidad de néctar de cada fuente de alimento. Los valores de la función objetivo sirven como guía, para que los diferentes tipos de abejas (operadores) evolucionen la población. Este proceso se repite hasta que se alcanza el criterio de parada. El desempeño del ABC ha sido comparado con otros métodos como son GA, DE y PSO [37 y 38]. Los resultados muestran que ABC produce soluciones óptimas y efectivas para diversos problemas de optimización. Tales características motivan que el ABC se utilice para resolver diferentes tipos de problemas en diferentes campos de la ingeniería [39-44]. Una ventaja relevante del algoritmo ABC es que no emplea una estrategia local para el cálculo de nuevas soluciones. En lugar de eso, se usa un conjunto de operadores aleatorios para construirlas, con esto se evita caer en algún óptimo local. Además, el uso de diversos tipos de abejas en lugar de sólo uno hace que el ABC sea atractivo para su implementación en diferentes áreas de ingeniería.

A pesar de que los AM han demostrado ser eficaces, existe una tendencia en la cual se propone la hibridación de técnicas con el objetivo de mejorar su rendimiento y velocidad. Dentro del área de aprendizaje automático, el mecanismo *Opposition-Based Learning* (OBL), inicialmente propuesto por Tizhoosh et al. [45]. Considera al mismo tiempo, la solución estimada actual y su valor opuesto correspondiente, para alcanzar una rápida aproximación de la solución candidata actual. Se ha comprobado matemáticamente [46-48], que en comparación con una solución candidata aleatoria, una solución candidata opuesta, tiene una mayor probabilidad de acercarse más rápidamente a una solución global óptima. Recientemente los conceptos de OBL se han usado para acelerar algunos algoritmos metaheurísticos, como son: GA [49], DE [50], PSO [51] y *Gravitational Search Algorithm* (GSA) [52 y 53].

Por otra parte, el procesamiento digital de señales (PDS) es una herramienta ampliamente utilizada en ingeniería electrónica e informática, ya que permite extraer información de sistemas físicos convirtiéndolas de señales analógicas en digitales. El tener una señal digital permite que sea manipulada y analizada por medio de algoritmos en un ordenador. Dichos algoritmos dependen de la aplicación que se esté realizando, pero existe un paso clave dentro del PDS, como es el filtrado. Que es un proceso en el que se busca eliminar el ruido y frecuencias no deseadas, que puedan tener las señales analógicas al ser obtenidas y transformadas a digitales [54]. Cabe señalar que las señales digitales pueden ser mono-dimensionales o multidimensionales, dependiendo del número de fuentes de las que provienen [54]. Sin embargo, los filtros digitales (FD) se aplican en ambos casos.

Los filtros digitales son clasificados en dos principales categorías [54]: *Finite Impulse Response* (FIR) e *Infinite Impulse Response* (IIR). La salida de un filtro FIR se calcula en función del valor actual y valores previos de la entrada. Esto implica que cuando no existe una señal de entrada, la salida tenderá a desaparecer. Por otro lado, la salida de un filtro IIR es dominada no sólo por el valor presente y previo de la entrada, sino también por los valores pasados de su propia salida. Esta realimentación produce el hecho de que cuando la entrada deja de existir, las salidas pasadas del filtro seguirán excitándolo. La identificación y modelado de filtros digitales IIR es de gran importancia en diversas áreas de la ingeniería, tales como procesamiento de señales, control, sistemas de comunicaciones, procesamiento de imágenes y eliminación de ruido [55]. El proceso de identificación consiste en alterar los coeficientes de un determinado filtro modelo, para estimar los parámetros de un sistema desconocido, a partir del conocimiento de su entrada y salida.

Para desempeñar esta tarea es necesario la minimización de una función objetivo. Típicamente el error cuadrático medio (*Mean Square Error*, MSE), existente entre la respuesta del sistema desconocido y la salida del filtro modelado.

Para identificar los coeficientes de los FD, se han desarrollado numerosos algoritmos que hacen uso de diversas técnicas metaheurísticas. Algunos ejemplos son: el trabajo realizado por Howell y Gordon en 2001 [55], en el cual se emplea un algoritmo de identificación basado en *Learning Automata* (LA). La propuesta de Kalinli y Karaboga del 2005 [56], que introduce el uso de *Artificial Immune Algorithm* (AIA) como técnica de optimización para el diseño de filtros IIR. Recientemente, existe el enfoque presentado por Rashedi en 2011 [52]. En el cual se propone el modelado de filtros digitales usando el *Gravitational Search Algorithm* (GSA). Otra propuesta, que además en la literatura se usa como forma de comparación estándar es la presentada en Lin et al. en 2008 [57]. En la cual se propone un algoritmo de identificación basado en el popular método *Particle Swarm Optimization* (PSO). A pesar de la gran aceptación que tienen estas propuestas, existen algunas deficiencias que afectan significativamente a su desempeño [58]. Las cuales se abordarán en las siguientes secciones de este trabajo de investigación.

Las señales digitales multidimensionales más comunes son las imágenes digitales. Por ejemplo, una imagen en escala de grises es una señal digital bidimensional, en la cual se cuantifican los niveles de intensidad de luz. Una imagen se representa normalmente como  $I(x, y)$ , donde el valor de la intensidad se obtiene por los índices de las coordenadas  $(x, y)$ . La forma más sencilla de representar una imagen es por medio de una matriz, de tal modo que cada elemento de la matriz corresponde a un valor de intensidad. Desde este punto de vista, las técnicas de PDS son aplicadas a las imágenes, sin embargo por su propia naturaleza, la comunidad científica ha creado una nueva vertiente llamada Procesamiento Digital de Imágenes (PDI). Desde sus inicios el PDI ha jugado un papel importante en el área de las ciencias computacionales. Esto se debe a que es posible adquirir diversas muestras de un ambiente específico, el cual puede ser controlado o no. El objetivo de tener diversas imágenes es poder analizarlas y extraer características importantes. De modo similar a como lo hace el ojo humano, un ejemplo de esto es la reconstrucción tridimensional de superficies o terrenos en el espacio 3D para navegación en robótica [59]. De este modo, los sistemas de PDI son pieza clave en diversos procesos de ingeniería. Dentro del análisis de imágenes existen diferentes etapas de procesamiento, las cuales permiten manipular la imagen y analizar los objetos que están presentes en cada escena.

Estas etapas comprenden desde la clasificación de los niveles de intensidad (segmentación), hasta tareas que tienen que ver con la identificación y detección de objetos. Los sistemas PDI deben ser capaces de adaptarse a las condiciones de cada imagen, es aquí donde los algoritmos inteligentes logran integrarse. Definir una función objetivo que cumpla con los requerimientos del problema, es uno de los puntos clave para poder aplicar los AM al procesamiento de imágenes.

Dentro del PDI una de las tareas clave es la segmentación, que consiste en clasificar los píxeles de la imagen de acuerdo a su valor de intensidad. La umbralización (*Thresholding*, TH, se usará para referirse a este proceso) es el método más sencillo de segmentación, el cual trabaja tomando un valor de un umbral (*threshold*) para clasificar los píxeles. Si el valor de intensidad de un píxel es mayor que el umbral ( $th$ ), entonces ese píxel es etiquetado en la primera clase, mientras que el resto corresponden a la segunda clase. La mayoría de las implementaciones de PDI propuestas en años recientes, requieren del uso de una metodología de TH. Es por esto que un gran número de técnicas han sido ampliamente estudiadas. De forma general, los métodos de TH se dividen en paramétricos y no-paramétricos [60-63]. Los enfoques paramétricos necesitan estimar valores de la función de densidad de probabilidad para modelar cada clase. El proceso de estimación consume demasiado tiempo, además de ser computacionalmente costoso. Por otra parte, los TH no-paramétricos emplean una gran cantidad de criterios para verificar la calidad del valor de umbral tales como la varianza entre clases, la entropía y el rango de error [64-66]. Estas métricas pueden también usarse como funciones a optimizar, ya que por su robustez y precisión son una opción atractiva. En la literatura referente, existen diversos métodos clásicos de umbralización [67]. Dos de ellos son dos referencias clásicas. El primero propuesto por Otsu en [64], maximiza la varianza entre las clases, mientras que el segundo método, presentado por Kapur et al. [65], usa la maximización de la entropía para medir la homogeneidad entre las clases. Se han propuesto diversos AM para encontrar los mejores umbrales, por ejemplo PSO o el método de ABC, los cuales han sido empleados para segmentar imágenes [68]. En ambos casos se emplea como función objetivo la entropía de Kapur et al. [65]. El *Bacterial Foraging Algorithm* (BFA) [69], ha sido implementado para encontrar los valores óptimos que den solución a las ecuaciones de Otsu y Kapur [70].

Detectar objetos en las imágenes, permite proveer información importante acerca del ambiente que se analiza a través de las tomas. Es posible decir que la detección de objetos en general constituye uno de los objetivos principales en PDI, debido a que es la

forma que más se aproxima al ojo humano. Las aplicaciones incluyen entre otros la supervisión, la inspección visual, la navegación de vehículos autónomos, el análisis de imágenes médicas, etc. Los AM han resultado ser útiles en un gran número de áreas de la informática [71], con relevantes contribuciones en optimización, reconocimiento de patrones, detección de formas y aprendizaje automático. En particular, han obtenido un interés de investigación considerable, por parte de la comunidad de visión por computador (VC), ya que son algoritmos inspirados en la naturaleza que han contribuido a la solución de complejos problemas dentro de los temas de visión, un ejemplo de este tipo de problemas es la detección de cambios entre imágenes [72].

Por otro lado, los glóbulos blancos (GB), también conocidos como leucocitos, juegan un rol importante en el diagnóstico de diferentes enfermedades. Las técnicas de procesamiento digital de imágenes, han contribuido exitosamente en la creación de nuevos métodos para el análisis de células, los cuales, a su vez, han dado lugar a un diagnóstico de enfermedades más preciso y fiable. Sin embargo, las grandes variaciones en las células con respecto a tamaño, forma, contorno y localización, complican el proceso de extracción de datos. Además, el contraste entre los límites de las células y el fondo de la imagen, puede verse alterado debido a que las condiciones de iluminación son inestables durante la captura de las imágenes. Numerosos trabajos se han desarrollado en el área de detección de células sanguíneas. En [73] se propone un método para identificar GB basado en Máquinas de Vectores Soporte, (*Support Vector Machines*, SVM). En tal enfoque, la intensidad de cada píxel se usa para construir vectores de características, mientras que una SVM se utiliza para clasificarlos y segmentarlos. Por medio del uso de un enfoque diferente, en Wu et al., [74] desarrollan un método de Otsu iterativo, para segmentación de leucocitos basado en el histograma circular. De acuerdo con esta técnica, las imágenes de frotis sanguíneo se procesan en el espacio de color descrito por matiz - saturación - intensidad (*Hue-Saturation-Intensity*, HSI), considerando que el componente de matiz (Hue) contiene la mayoría de la información de los GB. Uno de los últimos avances en la investigación sobre la detección de GB, es el algoritmo propuesto por Wang et al. [75], el cual está basado en (*Fuzzy Cellular Neural Network*, FCNN). A pesar de que este método se ha probado con éxito en la detección de un único leucocito, no hay evidencia de que pueda detectar varios. Además, su rendimiento comúnmente decae, cuando no se define apropiadamente el número de iteraciones, dando como resultado un problema complejo, ya que no se proporciona ninguna pista clara de cómo seleccionar apropiadamente dicho valor.

Dado que las células sanguíneas se pueden aproximar por formas cuasi-circulares, resulta útil el uso de un algoritmo detector de círculos para su identificación. El problema de la detección de características circulares tiene suma importancia para el análisis de la imagen, en particular en imágenes médicas [76]. La detección de círculos se realiza generalmente por medio de la transformada de Hough circular [77]. Un enfoque típico basado en Hough, emplea un detector de bordes, cuya información sirve de guía para localizar los círculos y sus respectivos valores de radio. La detección de estos puntos de interés, se realiza promediando, filtrando y analizando el histograma del espacio de transformación. Como alternativa a las técnicas basadas en la transformada de Hough (HT), el problema de detección de círculos ha sido también resuelto por medio del uso de métodos de optimización, los cuales en general, han proporcionado mejores resultados que los que emplean HT, en términos de precisión, velocidad y robustez [78]. Tales enfoques han generado varios detectores de círculos robustos, que usan diferentes algoritmos de optimización como son: GA [78], HSA [79], DE [80] y EMO [81].

Como se mencionó anteriormente, la localización y reconocimiento de objetos en imágenes digitales son campos de estudio importantes dentro de la visión por computador y procesamiento de imágenes. El método de comparación de plantillas (*Template Matching*, TM) es una técnica de PDI, que tiene como objetivo localizar y reconocer objetos en tomas extraídas de diversos ambientes, mediante la similitud existente entre una sub-imagen, mejor conocida como plantilla y una región coincidente en la imagen fuente. La idea general de TM involucra dos puntos críticos: la medición de la similitud y la estrategia de búsqueda [82]. Para el primer punto, existen diversas métricas que son empleadas para evaluar la similitud entre dos imágenes, las más importantes son la suma de diferencias absolutas (SAD), la suma de diferencias cuadráticas (SSD) y la correlación cruzada normalizada (NCC), siendo esta última la más usada en la literatura. El cálculo de estas métricas es la tarea más costosa desde el punto de vista computacional y representa la operación que más tiempo consume en el proceso de TM [83]. Por otro lado, la estrategia de búsqueda más sencilla en TM es el algoritmo de búsqueda completa, que realiza una detección óptima del objeto en cuestión, calculando la métrica seleccionada para cada uno de los píxeles de la imagen fuente. Por desgracia, la búsqueda exhaustiva en todos los puntos de la imagen es extremadamente costosa en términos computacionales. Recientemente, se han propuesto diversos algoritmos basados en TM y enfoques evolutivos. El propósito de estas implementaciones es reducir el número de evaluaciones de la métrica seleccionada (usada como función objetivo), calculando sólo para un subconjunto de

localizaciones de búsqueda. Los enfoques propuestos en la literatura son detectores robustos que hacen uso de diferentes técnicas de optimización como son: algoritmos genéticos [84], optimización por enjambre de partículas [85, 86] y el algoritmo imperialista competitivo (*imperialist competitive algorithm*, ICA) [87]. A pesar de que estos métodos reducen significativamente la cantidad de localizaciones de búsqueda, sufren de convergencia prematura, lo cual genera problemas que se abordarán en los siguientes capítulos del presente documento.

Continuando con las aplicaciones que tienen los AM en ingeniería, existen diversas áreas que por su naturaleza son consideradas prioritarias y actualmente están acaparando la atención de la comunidad científica. El objetivo de las investigaciones en dichos temas, es dar solución a los desafíos que contribuyen al beneficio de la humanidad. Un ejemplo de esto es la crisis energética, que se ha incrementado en años recientes a nivel mundial, afectando directamente los precios de las fuentes tradicionales de energía que en su mayoría se basan en combustibles fósiles. Además, el uso de las fuentes tradicionales de energía, incrementa de forma substancial los niveles de contaminación. Con el objetivo de preservar el medio ambiente y como ayuda a la recuperación de la economía a nivel mundial, es indispensable la búsqueda de fuentes de energía alternativa y renovable. La energía solar (ES) es una interesante opción, que ha llamado la atención de los investigadores, debido a que durante todo el año existen muchos días soleados en distintas áreas del planeta. La principal aplicación de la ES, es la provisión de energía eléctrica que pueda usarse en la industria y el hogar. El primer uso de la ES, fue a finales de la década de 1950 en sistemas espaciales (vehículos y satélites) y fue elegida por el bajo mantenimiento que requiere [88, 89]. Durante años, la forma de generar electricidad a partir del sol ha sido mediante el uso de celdas fotovoltaicas (CF) o celdas solares (CS), habiéndose incrementado su aplicación masivamente en los últimos 10 años [89-90]. Debido a esto el diseño de CF o CS, se ha expandido considerablemente. El propósito del diseño de CF es obtener la mayor potencia eléctrica a partir de la energía solar. Para esto existen dos pasos fundamentales; el primero, consiste en formular el modelo matemático y el segundo, la estimación de los parámetros de la CS. Se han propuesto una gran cantidad de representaciones matemáticas, bajo diferentes condiciones [92-97]. Sin embargo en términos prácticos se emplea el uso de dos circuitos electrónicos equivalentes, conocidos como los modelos de diodo simple (DS) y de doble diodo (DD) [98]. Sin importar el modelo seleccionado, existen parámetros que deben ser estimados o identificados, los cuales afectan directamente en la eficiencia de la CS. Los métodos empleados, para identificar los parámetros de los modelos basados en diodos, se

pueden dividirse en dos grupos o métodos: deterministas o analíticos y metaheurísticos. Dentro del grupo de los deterministas, se han propuesto varios enfoques, por ejemplo, el uso de una técnica basada en la ley de Newton en combinación con una modificación de la técnica de mínimos cuadrados [99]. Otro aporte interesante, es el conocido como de la función co-contenida, que emplea la función de Lambert para extraer los mejores parámetros de las celdas solares [100]. Asimismo, se realizó un estudio comparativo de tres técnicas de estimación de parámetros de SC, en ellas se incluyen el ajuste de curvas y los métodos iterativos de tres y cinco puntos [101]. Por otro lado, con el incremento de la potencia de cómputo ha sido posible el uso de métodos alternativos para el problema de SC. Los GA clásicos han sido usados para estimar los parámetros de los modelos de diodos [102]. AlRashid et al. [89] han propuesto una modificación de los GA [89], donde se emplean patrones de búsqueda predefinidos para mejorar el rendimiento de los GA. Otro AM usado para estos fines es el PSO [97] y una modificación del mismo propuesta por Wei et al. [103].

## 1.2 Planteamiento del problema

Los métodos planteados en la sección 1.1 han sido probados ampliamente y sus resultados son muestra de las capacidades para encontrar soluciones a las aplicaciones propuestas. Sin embargo, tras un análisis profundo, se han encontrado deficiencias, las cuales se han abordado en este trabajo para tratar de resolverlas. Esta investigación se divide en cuatro partes, en las cuales se puede comprobar la eficiencia y utilidad de las técnicas metaheurísticas, para resolver problemas. Concretamente, en las áreas de procesamiento de señales e imágenes digitales, así como, en aplicaciones a problemas seleccionados de ingeniería, como son las energías alternativas. De este modo, a continuación se describen los problemas a tratar en cada uno de los temas antes mencionados.

*Algoritmos metaheurísticos:* existen numerosas pruebas que corroboran la eficiencia de los algoritmos de optimización metaheurísticos. Sin importar la metáfora en la que estén basados, estos métodos deben cumplir con un balance entre los procesos de exploración y explotación. Sin embargo, dependiendo del problema que se esté resolviendo, algunas técnicas metaheurísticas se centran más en la exploración que en la explotación, incrementando el costo computacional. Tal es el caso del algoritmo EMO, que pese a su gran desempeño presenta algunas deficiencias. EMO tiene cuatro etapas, inicialización,



búsqueda local, cálculo de fuerzas y movimiento. La búsqueda local (BL) es un proceso estocástico que analiza diferentes regiones en la vecindad de cada partícula de la población. El principal inconveniente de EMO, es su complejidad computacional, resultante del gran número de iteraciones que comúnmente se requieren durante el proceso de BL. Esta situación empeora cuando el número de dimensiones del problema a optimizar se incrementa. Se han propuesto en la literatura algunos enfoques para reducir el costo computacional de EMO. Guan et al. [104] han propuesto la codificación del grupo de partículas, con el objetivo de reducir las direcciones de búsqueda en cada dimensión. En [105, 106], los autores incluyen un nuevo método de BL, basado en patrones de búsqueda fijos y una estrategia de reducción, que acorta el tamaño de la población a lo largo de las iteraciones. Otra propuesta adopta una BL modificada que emplea el gradiente descendiente para mejorar la complejidad computacional [14]. Estos enfoques han mejorado el tiempo de cómputo requerido por el algoritmo EMO original. Algunos trabajos recientes [23,107] han demostrado que las modificaciones o reducciones en el proceso de búsqueda local, afectan a las propiedades de convergencia, exploración, precisión y a la diversidad de la población. En base a lo anterior, dentro de los algoritmos metaheurísticos, el problema a tratar es el alto costo computacional del EMO original, generado en la fase de búsqueda local. Las soluciones propuestas no deben repercutir en la eficiencia del EMO original.

*Diseño de filtros digitales:* sin importar de qué tipo de filtro se trate (FIR o IIR), es necesario modificar sus coeficientes para obtener una salida deseada. Como ya se mencionó, para verificar la calidad del FD de acuerdo a un determinado grupo de coeficientes, se emplea el MSE. En la investigación desarrollada sólo se tratan los filtros IIR debido a su importancia. Aun así y teniendo en cuenta la amplia gama de aplicaciones de los filtros IIR, su identificación y modelado se considera un proceso complejo que requiere de algoritmos de optimización globales capaces de operar sobre superficies de error multimodales típicas de estos procesos. En la sección 1.1 se mencionan implementaciones en las cuales se hace uso de los algoritmos metaheurísticos. Sin embargo, Shilane et al. [108] en su trabajo del 2008, demuestran que existen algunas deficiencias que afectan significativamente su desempeño. Entre ellas se destacan la necesidad de un gran número de iteraciones y de elementos de la población para encontrar una solución lo más parecida al mínimo global, repercutiendo directamente en el costo computacional del algoritmo [109]. Además, para la mayoría de estos métodos, la convergencia no está suficientemente comprobada, por lo tanto no existe garantía de que los valores encontrados como solución, puedan ser considerados como los mejores. En este sentido, el problema

dentro del diseño de filtros está claramente definido: encontrar e implementar un algoritmo de optimización el cual sea capaz de modelar con alta precisión los filtros IIR, sin que esto repercuta en un alto costo computacional.

*Procesamiento de Imágenes:* como se mencionó anteriormente, la segmentación constituye una de las tareas más importantes dentro del procesamiento de imágenes, siendo por ello uno de los temas más estudiados en la literatura. En este trabajo se analizan los enfoques no-paramétricos, específicamente se estudian las métricas propuestas por Otsu y Kapur, debido a su gran popularidad. Como también se mencionó en la sección 1.1, las métricas allí descritas pueden utilizarse como funciones objetivo. En este caso, su uso representa un proceso iterativo, en el cual se busca encontrar los mejores valores que maximicen la función elegida. La eficiencia y precisión de ambas propuestas, han sido comprobadas para la segmentación bi-nivel (un umbral y dos clases) [62]. Aunque pueden expandirse para más de dos clases, su costo computacional se incrementa exponencialmente cuando se incorpora un nuevo umbral [63]. A pesar de que se ha propuesto el uso de diversos algoritmos metaheurísticos, para encontrar múltiples umbrales, sus resultados carecen de precisión, robustez y el costo computacional que representan continúa siendo elevado. Ejemplo de esto es el análisis realizado por Hammouche et al. [110], donde se describe la aplicación de diferentes algoritmos para resolver las ecuaciones de Otsu y Kapur. Los resultados mostrados en este estudio demuestran la eficiencia de los algoritmos. Sin embargo también muestran las deficiencias antes mencionadas. De este modo, los problemas a resolver dentro de la segmentación de imágenes se encaminan a encontrar un método que sea capaz de optimizar las funciones objetivo de Otsu y Kapur, para encontrar múltiples umbrales. Dicha técnica debe ser competitiva, robusta y ofrecer resultados precisos sin incrementar su costo computacional.

Por otra parte, en la sección 1.1, dentro del tema de imágenes médicas y específicamente en lo que respecta a la detección de GB, se mencionó que es posible llevar a cabo esta tarea mediante el uso de la transformada de Hough circular. Sin embargo, este enfoque requiere de un gran espacio de almacenamiento debido a las celdas tridimensionales que cubren todos los parámetros de los círculos, posición de centro  $(x, y)$  y tamaño del radio  $(r)$ . Lo que también implica una alta complejidad computacional produciendo una baja velocidad de procesamiento. La precisión de los parámetros extraídos para el círculo detectado es pobre, particularmente en presencia de ruido [111]. Para una imagen digital con un tamaño significativamente grande y un mapa de píxeles de borde

densamente poblado, el tiempo de procesamiento requerido por la transformada de Hough circular, hace que sea prohibitiva su implementación en aplicaciones en tiempo real. Con el fin de superar tal problema, algunos otros investigadores han propuesto nuevos enfoques basados en la HT, por ejemplo la transformada de Hough probabilística [112, 113], la HT aleatoria [114] y la Fuzzy HT [115]. También se han propuesto transformaciones alternativas, como la propuesta por Becker et al. [116]. A pesar de que estos nuevos enfoques han demostrado mayor velocidad de procesamiento en comparación con la HT original, aún continúan siendo sumamente sensibles al ruido. Por su parte, los algoritmos de detección basados en enfoques de optimización, presentan diversas ventajas en comparación con aquellos basados en la HT; además, su aplicación para detectar glóbulos blancos es escasa. Una excepción es el trabajo presentado por Karkavitsas y Rangoussi [76] que resuelve el problema de detección de GB por medio del uso de GA. Sin embargo, ya que la función objetivo mide la calidad de cada solución y sólo considera los píxeles contenidos dentro de un círculo con radio fijo, el método es propenso a producir falsas detecciones, en particular para imágenes que contienen GB irregulares o traslapados. De este modo, los problemas existentes en relación a la detección de células sanguíneas por medio de formas circulares, derivan en el hecho de que los métodos a desarrollar deben cumplir una serie de requisitos: ser robustos, capaces de interactuar con células en condiciones complejas y que el tiempo de cómputo sea menor que el de las propuestas que emplean la HT.

Para la tarea de localización y reconocimiento de objetos en imágenes se hace uso del método de comparación de plantillas y como métrica de similitud se emplea la correlación cruzada normalizada (NCC). Sin embargo, ya que la búsqueda exhaustiva es un proceso que consume recursos de cómputo excesivos, se hace uso de las herramientas metaheurísticas, que consideran a la NCC como función objetivo. El objetivo principal de emplear técnicas de búsqueda metaheurísticas es reducir el número de evaluaciones de NCC, evaluando sólo un subconjunto de localizaciones. Sin embargo, una de las principales dificultades que se presentan es que TM resulta ser un problema de optimización discreto y los algoritmos metaheurísticos tienden a evaluar en múltiples ocasiones la misma solución candidata. En optimización discreta, el espacio de búsqueda está compuesto por un conjunto finito de soluciones, las cuales implican el uso de números aleatorios para el cálculo de nuevos individuos, entonces se pueden encontrar las mismas soluciones que han sido visitadas por otros individuos en iteraciones previas, esto ocurre particularmente cuando los individuos son confinados en áreas pequeñas [117]. Resulta evidente el hecho de

que las retracciones requeridas por el método incrementan el costo computacional. De este modo, dentro de la localización y reconocimiento de objetos, el problema principal es encontrar e implementar un algoritmo de búsqueda que además de reducir el número de evaluaciones de NCC, sea capaz de operar eficientemente en problemas de optimización discreta, ofreciendo una buena relación entre el consumo de recursos de cómputo y la precisión de los resultados.

*Energías alternativas:* dentro del campo de energía solar, el tema a tratar es la estimación de parámetros para los modelos de celdas fotovoltaicas. Los modelos de diodo simple y doble, requieren de una estimación de cinco y siete parámetros respectivamente. Dado que la eficiencia de la CS depende de dichos parámetros, el principal problema es identificar sus valores, para que, al ser aplicados en el modelo elegido, produzcan salidas que se asemejen a las mediciones experimentales [98]. Como se mencionó en el apartado 1.1, se han implementado métodos tradicionales para el diseño de las CS. Sin embargo, requieren que el problema cumpla con algunas características matemáticas, como son: continuidad, convexidad y diferenciabilidad. Esto significa, que es necesario tener algunas consideraciones previas antes de iniciar el proceso de estimación. Lo cual representa un problema complejo, requiriendo de un experto humano para realizar los ajustes. Además del costo computacional que conlleva en su ejecución, son altamente susceptibles para el proceso de inicialización, siendo posible afirmar que el éxito o fracaso, depende de este paso [98, 118]. Los resultados que se obtienen a partir de estas soluciones, carecen de precisión. Un problema importante en la estimación de parámetros de CS, es que la función objetivo es altamente no-lineal [118]. Por otro lado, se ha propuesto el uso de AM para el problema de las celdas solares. Estos enfoques proveen alternativas al uso de los métodos deterministas, ya que no requieren ninguna configuración previa. Sin embargo, debido a la no-linealidad del problema, es complicado encontrar soluciones exactas en cada ejecución. Resumiendo, para el campo de energía solar, el problema consiste en encontrar un AM capaz de estimar los valores de las celdas solares con exactitud, garantizando que los resultados van a repetirse cada vez que se ejecute y también debe ser adaptable a las modificaciones del modelo de diodo a estimar.

### 1.3 Objetivos

Tras el análisis de los antecedentes y problemas identificados en cada tema a tratar, se definen los objetivos de esta investigación, los cuales son de doble naturaleza, generales

y específicos.

### 1.3.1 *Objetivos generales*

1. Adaptar algoritmos metaheurísticos para mejorar su rendimiento.
2. Desarrollar métodos capaces de resolver problemas de procesamiento de señales e imágenes digitales, así como también en el área de energías alternativas empleando algoritmos metaheurísticos.

### 1.3.2 *Objetivos específicos*

1. Realizar una modificación al algoritmo EMO estándar usando la estrategia *opposition-based learning*, con la finalidad de reducir el costo computacional que se requiere en la búsqueda local sin demeritar las capacidades del algoritmo EMO original [119].
2. Presentar una nueva propuesta para la identificación y modelado de filtros digitales IIR basada en el algoritmo EMO [120].
3. Proponer un nuevo algoritmo de segmentación multinivel para imágenes digitales, que combine las características del HSA original con las metodologías propuestas por Otsu y Kapur [121].
4. Implementar el algoritmo EMO como detector de círculos para la identificación de células sanguíneas en imágenes médicas [122].
5. Proponer una nueva función objetivo que mejore la precisión y robustez en la detección de células sanguíneas, dicha función se emplea en el detector propuesto en el objetivo específico 3 [123].
6. Desarrollar un método para la localización y reconocimiento de objetos en imágenes digitales usando la técnica de comparación de plantillas y una modificación del algoritmo EMO [124].
7. Crear un algoritmo para el diseño de celdas solares, mediante la aplicación del método ABC para estimar los parámetros de los modelos basados en diodos [125].
8. Realizar los experimentos pertinentes para los objetivos 1-7.
9. Analizar los resultados experimentales del objetivo específico 8 y compararlos con algunos otros métodos existentes en la literatura.
10. Divulgar y difundir los resultados de la investigación mediante publicaciones científicas u otros medios [119 - 126].

## 1.4 Metodología

La presente investigación se ha realizado siguiendo las directrices marcadas por el

método científico. De forma más concreta, inicialmente se ha realizado una búsqueda de información bibliográfica en todas las fuentes disponibles, utilizando principalmente para ello el acceso a través de la red del Consorcio Madroño al que están suscritas las Universidades de la Comunidad de Madrid, y en concreto la Universidad Complutense. Se han estudiado los antecedentes en las áreas mencionadas. Posteriormente, se han desarrollado las etapas de trabajo que se indican a continuación, manteniendo actualizadas las referencias bibliográficas mediante una revisión continua de las mismas. Cabe señalar que este trabajo de investigación se ha desarrollado en conjunto con el Centro Universitario de Ciencias Exactas e Ingenierías de la Universidad de Guadalajara en México.

Las etapas de desarrollo seguidas han sido:

1. Definición de los problemas a tratar mediante el uso de algoritmos metaheurísticos, que constituyen la motivación y objeto de la investigación.
2. Planteamiento de los objetivos de la investigación.
3. Diseño y desarrollo de la metodología.
4. Obtención de resultados experimentales.
5. Clasificación, análisis e interpretación de los datos.
6. Conclusiones generales.
7. Descripción del trabajo futuro.

Para cada problema a tratar se han diseñado un conjunto independiente de experimentos basados en las técnicas propuestas. Este proceso incluye, adquisición de datos, procesamiento de la información, desarrollo e implementación de los diversos algoritmos de optimización, obtención de los resultados experimentales y finalmente la validación de los mismos.

## 1.5 Aportaciones de la investigación

En el desarrollo de la investigación se ha logrado dar solución a los problemas planteados. Tras analizar los resultados experimentales y con el objetivo de presentar alternativas que en la comunidad científica sirvan de guía para futuros trabajos, se han logrado realizar las siguientes publicaciones:

**Artículos publicados en revistas indexadas en el *Journal Citation Reports* (JCR):**

**Autores:** Erik Cuevas, Diego Oliva, Daniel Zaldívar, Marco Pérez-Cisneros y Gonzalo Pajares.

**Título:** “Opposition-Based Electromagnetism-Like for Global Optimization”.

**Revista:** International Journal of Innovative Computing Information and Control.

**Vol, pag. (año):** 8, 12, 8181-8198, 2012.

Este artículo presenta una mejora al algoritmo *Electromagnetism – like Optimization* (EMO), en la cual se emplea el enfoque de *Opposition - Based Learning* (OBL) para acelerar la velocidad de convergencia global. El método OBEMO propuesto reduce significativamente el costo computacional requerido sin afectar o deteriorar las habilidades del algoritmo EMO original. Los experimentos realizados sobre un grupo considerable de funciones de prueba, muestran que el OBEMO proporciona resultados de rendimiento prometedores para la mayoría de los problemas analizados. Este método se describe en el capítulo tres.

**Autores:** Erik Cuevas, Diego Oliva, Margarita Díaz, Daniel Zaldívar, Marco Pérez-Cisneros y Gonzalo Pajares.

**Título:** “White Blood Cell Segmentation by Circle Detection Using Electromagnetism-Like Optimization”.

**Revista:** Computational and Mathematical Methods in Medicine.

**Número, año:** 395071, 2013.

Este artículo presenta un algoritmo para la detección automática de glóbulos blancos, incrustados dentro de imágenes complicadas, provenientes de muestras de frotis sanguíneos. El problema de detección de glóbulos blancos se considera un problema de detección de formas circulares. Este enfoque está basado en el algoritmo EMO. Para el proceso de detección se propone el uso de una función objetivo la cual mide la similitud del círculo candidato con el actual GB. Guiado por los valores de esta función objetivo, un grupo de círculos codificados evolucionan mediante la aplicación de los operadores EMO, dichos círculos se ajustan sobre las células sanguíneas contenidas dentro del mapa de bordes de la imagen. Los resultados experimentales provenientes de imágenes de células sanguíneas con diferentes rangos de complejidad se incluyen para validar la eficiencia de la técnica propuesta, con respecto a la detección, robustez y estabilidad. Esta técnica se describe completamente en el capítulo tres.

**Autores:** Diego Oliva, Erik Cuevas, Gonzalo Pajares, and Daniel Zaldívar.

**Título:** “Template Matching using an improved Electromagnetism-Like algorithm”.

**Revista:** Journal of Applied Intelligence.

**Vol, pag. (año):** 41, 3, 791-807, 2014.

En este artículo se introduce una modificación del algoritmo EMO que en combinación con la técnica de comparación de plantillas, es capaz de detectar objetos seleccionados en imágenes digitales. Dicha modificación se realiza en la búsqueda local implícita en el algoritmo EMO. El método EMO propuesto usa como función objetivo la correlación cruzada normalizada (NCC) para identificar los objetos en las imágenes médicas. Para este proceso se requiere como paso previo la selección de una plantilla en la imagen, de suerte que la técnica propuesta realizará la búsqueda haciendo comparaciones por medio del NCC y el algoritmo EMO. Este procedimiento se presenta en el capítulo tres.

**Autores** Diego Oliva, Erik Cuevas, Gonzalo Pajares, Daniel Zaldívar y Marco Pérez-Cisneros.

**Título:** “Multilevel thresholding segmentation based on harmony search optimization”.

**Revista:** Journal of Applied Mathematics.

**Número, año** 575414, 2013.

En este artículo se presenta un algoritmo de segmentación multinivel (MT) basado en *Harmony Search* (HSA). Este enfoque combina las habilidades de búsqueda de HSA, con las funciones objetivo propuestas en los métodos populares de segmentación de Otsu y Kapur. La combinación de estas técnicas, genera un algoritmo de segmentación multinivel, que puede identificar efectivamente los valores de umbral de una imagen digital en un número reducido de iteraciones. Los resultados experimentales, muestran el alto rendimiento de este método para la segmentación de imágenes digitales. El algoritmo propuesto se presenta en el capítulo cuatro.

**Autores:** Diego Oliva, Erik Cuevas and Gonzalo Pajares.

**Título:** “Parameter identification of solar cells using artificial bee colony optimization”.



**Revista:** Energy Journal.

**Vol, pag. (año):** 72, 1, 75-82, 2014.

En este artículo se propone la implementación del algoritmo de optimización *Artificial Bee Colony* (ABC), para la identificación y modelado de celdas solares. El modelo de celdas solares que se emplea está basado en diodos y el algoritmo ABC se encarga de encontrar los parámetros que den una respuesta eficiente en relación al voltaje-corriente, existente en la salida del circuito que modela la celda solar. Como función objetivo se emplea el error cuadrático medio (RMSE), para esto se emplean valores experimentales, los cuales pueden ser obtenidos por medio de mediciones o proporcionados por el fabricante. El método propuesto, es capaz de encontrar los parámetros que aplicados al circuito propuesto, den como salida valores cercanos a las muestras experimentales. La implementación de este método se detalla en el capítulo cinco.

**Autores:** Diego Oliva, Erik Cuevas, Gonzalo Pajares, Daniel Zaldívar and Valentín Osuna.

**Título:** “A Multilevel Thresholding algorithm using electromagnetism optimization”.

**Revista:** Neurocomputing.

**Vol, pag. (año):** 139, 357-381, 2014.

En este artículo se presenta un algoritmo de segmentación multinivel (MT) basado en el EMO. Este método combina las habilidades de búsqueda de EMO, con las funciones objetivo propuestas en los métodos populares de segmentación de Otsu y Kapur. La combinación de estas técnicas, genera un algoritmo de segmentación multinivel, que puede identificar efectivamente los valores de umbral de una imagen digital en un número reducido de iteraciones. Los resultados experimentales, muestran el alto rendimiento de este método para la segmentación de imágenes digitales. La implementación de este método se es similar a la que se presenta en el capítulo cuatro correspondiente a [121].

**Autores:** Erik Cuevas, Daniel Zaldívar, Marco Pérez-Cisneros and Diego Oliva.

**Título:** “Block-matching algorithm based on differential evolution for motion estimation”.

**Revista:** Engineering applications of artificial intelligence.

**Vol, pag. (año):** 26, 488-498, 2013.

En este artículo se propone un algoritmo basado en evolución diferencial (DE) y en la técnica de ajuste de bloques (BM) para la estimación de movimiento. El uso de DE ayuda a reducir el número de localizaciones de búsqueda en el proceso de BM. El algoritmo estima los valores para algunas localizaciones, de la suma absoluta de diferencias (SAD), la cual es también considerada función objetivo. Para realizar la estimación se usan los valores de SAD previos calculados para las posiciones vecinas. Este método no considera patrones de búsqueda establecidos o alguna otra calibración previa, con esto se logra encontrar el óptimo real que corresponde al vector de movimiento más preciso. A pesar de que este método no se describe en la tesis ha servido de base para el desarrollo del trabajo presentado en [123].

**Artículos publicados en revistas no indexadas al JCR:**

**Autores:** Erik Cuevas, Diego Oliva.

**Título:** “Identificación y modelado de Filtros IIR usando un algoritmo inspirado en el Electromagnetismo”.

**Revista:** Ingeniería, Investigación y Tecnología.

**Vol, pag. (año):** 14, 1, 125-138, 2013.

En este artículo se propone un método de identificación y modelado de filtros de respuesta infinita al impulso (IIR), basado en el método de optimización EMO. Los resultados de simulación demuestran, que el enfoque propuesto tiene una mejor precisión y convergencia que los obtenidos por métodos reportados en la literatura para desempeñar la misma tarea. La descripción del enfoque propuesto se detalla en el capítulo tres.

## 1.6 Descripción de los capítulos

El presente trabajo de investigación se divide en seis capítulos, los cuales están organizados de acuerdo al desarrollo de la investigación. El orden en que lleva cada uno de ellos es el siguiente:

- **Capítulo 1:** describe la motivación y los objetivos que dieron origen a la investigación, de forma similar, se plantean los problemas a tratar y las aportaciones que han sido publicadas en revistas científicas de las áreas relacionadas.
- **Capítulo 2:** presenta el estado del arte de la investigación, aquí se hace un análisis de

las aportaciones más recientes dentro del tema de algoritmos metaheurísticos y sus aplicaciones en diversas áreas de ingeniería. Del mismo modo, se presenta el marco teórico, en el cual se introducen los temas de optimización global, filtrado digital de señales, procesamiento digital de imágenes y diseño de celdas solares. Remarcando el hecho de que cada una de las áreas de implementación debe ser considerada como un problema de optimización y que éste puede ser resuelto mediante el uso de algoritmos metaheurísticos.

- **Capítulo 3:** describe el algoritmo *Electromagnetism - like Optimization*, sus aplicaciones y una modificación del mismo. Además de la descripción de EMO, se explica la implementación realizada usando OBL, se detalla la aplicación de EMO para el diseño de filtros digitales, se explica su uso para detectar células sanguíneas en imágenes médicas y se plantea su implementación para detección de objetos por medio del uso de plantillas. Para cada caso, se realizan estudios comparativos entre los algoritmos propuestos y alternativas encontradas en la literatura, aplicados sobre problemas de prueba de cada área de estudio.
- **Capítulo 4:** aborda el tema de segmentación de imágenes digitales usando para esto el HSA. Se introduce primeramente el algoritmo HSA y se proporciona una descripción de los métodos de Otsu y Kapur para segmentación de imágenes. Se proporciona una explicación sobre el algoritmo de segmentación propuesto y se presenta un análisis de los resultados experimentales, obtenidos tras probar el método propuesto en un conjunto de imágenes de test.
- **Capítulo 5:** analiza el diseño de celdas solares, desde el punto de vista de la estimación de parámetros para un modelo basado en diodos, mediante el uso del algoritmo ABC. Define los conceptos del ABC y se proporciona un panorama general del problema de estimación de parámetros de las CS. Además de esto, se da una explicación del algoritmo propuesto para tal problema y se discuten los resultados experimentales y comparaciones realizadas.
- **Capítulo 6:** expone las conclusiones y plantea el trabajo futuro de la investigación

## **1.7 Resumen**

En este capítulo se presentan los antecedentes que dieron origen a la solución de los problemas planteados en la investigación. De forma similar se describe los objetivos, tanto el general como lo específicos y la metodología a seguir. Así mismo también son presentadas las aportaciones científicas de la investigación.



# Capítulo 2

## Marco teórico y estado del arte

---

### 2.1 Visión general

Este capítulo presenta una revisión de la bibliografía referente a los objetivos planteados en el capítulo uno. Allí se presentaron tres objetivos (diseño de filtros digitales, procesamiento de imágenes y energías alternativas) bajo el planteamiento de métodos basados en optimización. Pese a que la literatura presenta una gran cantidad de contenido de interés, se ha seleccionado sólo lo más destacado en cuanto a teoría y desarrollo de las áreas de investigación. Los aspectos tratados a lo largo de este capítulo se estructuran como se expresa a continuación, considerando en primer lugar dos enfoques de optimización, para seguidamente plantear los aspectos relacionados con los objetivos mencionados:

1. En primer lugar, se lleva a cabo un recorrido por la literatura, para encontrar los

fundamentos de la optimización mediante el uso de técnicas metaheurísticas.

2. En segundo lugar, se explican los conceptos de Aprendizaje basado en oposición (*Opposition-based Learning*, OBL) y su implementación como un método de optimización.
3. A continuación se exponen los principios del procesamiento digital de señales, se estudian los filtros digitales y su diseño, haciendo énfasis en las implementaciones realizadas desde el punto de vista de la optimización.
4. En cuarto lugar se realiza un análisis sobre conceptos de procesamiento de imágenes digitales, referentes a segmentación por umbralización, detección de células sanguíneas en imágenes médicas así como localización y detección de objetos.
5. Por último, se plantea un estudio de las técnicas de estimación de parámetros para el diseño de celdas solares, poniendo especial interés en las propuestas que hacen uso de técnicas de optimización.

## 2.2 Optimización y algoritmos metaheurísticos

La optimización se encuentra presente en prácticamente todas las áreas de estudio, es posible encontrar aplicaciones que van desde la economía a la medicina, pasando por la ingeniería y las ciencias aplicadas. Por sencillo que parezca, el problema de optimización presenta diversas complicaciones, es por esto que actualmente existen laboratorios y grupos de investigación dedicados a generar soluciones para las diversas aplicaciones.

### 2.2.1 Optimización

Cada problema a resolver tiene su propia forma de ser modelado, de forma genérica un problema de optimización se define de la siguiente manera [127]:

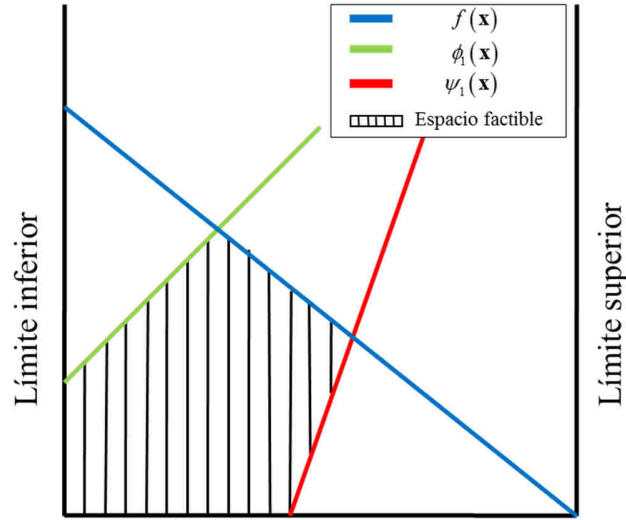
$$\begin{aligned} \underset{\mathbf{x} \in \mathcal{R}^n}{\text{Minimizar}} \quad & f_i(\mathbf{x}), \quad i = 1, 2, \dots, M, \\ \text{sujeto a} \quad & \phi_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, J, \\ & \psi_k(\mathbf{x}) \leq 0, \quad k = 1, 2, \dots, K, \end{aligned} \quad (2.1)$$

donde  $f_i(\mathbf{x})$ ,  $\phi_j(\mathbf{x})$  y  $\psi_k(\mathbf{x})$  son funciones del vector de diseño:

$$\mathbf{x} = \{x_1, x_2, \dots, x_n\}^T \quad (2.2)$$

Las componentes  $x_i$  de  $\mathbf{x}$  se denominan variables de decisión o diseño y pueden ser reales continuos, discretos o una mezcla de ambos. Las funciones  $f_i(\mathbf{x})$  se conocen

como funciones objetivo aunque en la literatura son a menudo llamadas funciones de costo o funciones de energía. El espacio cubierto por las variables de decisión se denomina espacio de búsqueda  $\mathcal{R}^n$ , mientras que el espacio formado por los valores de la función objetivo, se conoce como espacio de la solución (Figura 2.1).

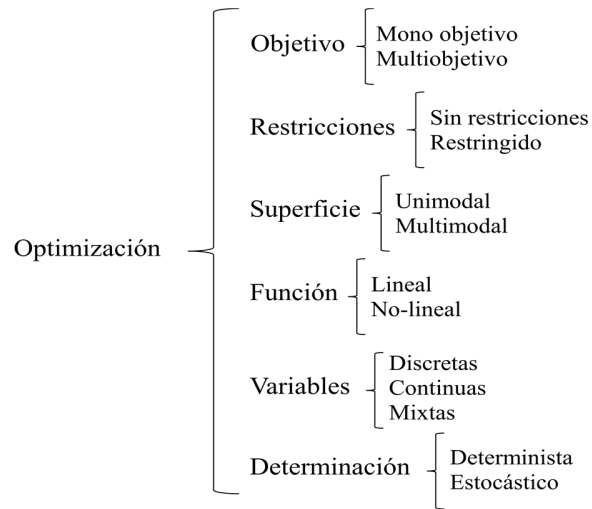


**Figura 2.1:** Espacio de búsqueda factible.

Las funciones objetivo pueden ser lineales o no lineales. Las igualdades para  $\phi_j$  y las desigualdades para  $\psi_k$  se denominan restricciones. Merece la pena señalar que también se pueden escribir las desigualdades de la forma  $\geq 0$  y se pueden formular los objetivos como un problema de maximización. Esto es debido a que la maximización de  $f_i(\mathbf{x})$  es equivalente a la minimización de  $-f_i(\mathbf{x})$  y cualquier desigualdad de  $g(\mathbf{x}) \leq 0$  es equivalente a  $-g(\mathbf{x}) \geq 0$ . Para las restricciones, el caso más sencillo para una variable de decisión  $x_i$  es  $x_{i,\min} \leq x_i \leq x_{i,\max}$ , conocido como límite. Si todas las restricciones  $\phi_j$  y  $\psi_k$  son lineales, entonces se dice que se trata de un problema linealmente restringido. Si tanto las restricciones como las funciones objetivo son todas lineales, es un problema de solución lineal.

Una clase especial de optimización sucede cuando no existen restricciones en absoluto (es decir,  $J = K = 0$ ) y la única tarea consiste en encontrar el valor óptimo de una única función objetivo. En este caso el problema se convierte en uno sin restricciones. Es posible clasificar los problemas de optimización de la siguiente manera:



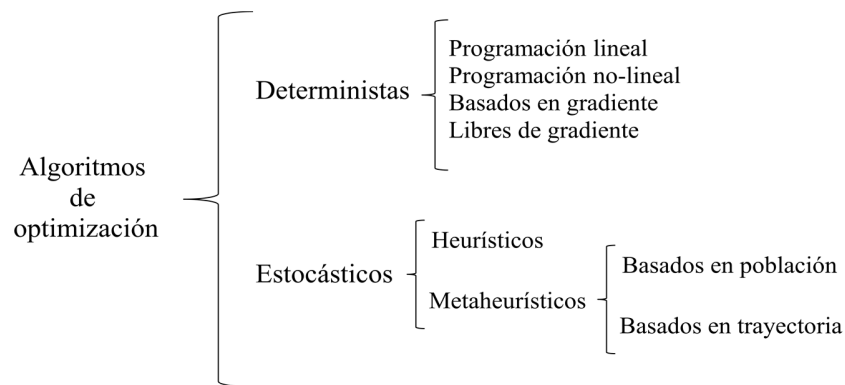


**Figura 2.2:** Clasificación de los problemas de optimización.

Existen diversos métodos estadísticos y matemáticos que se consideran como una forma distinta de optimización. Ejemplo de esto es el uso de los métodos de mínimos cuadrados, en procesamiento de datos, aquí se trata de minimizar la suma de los residuos de las diferencias entre los valores obtenidos por los modelos matemáticos y los valores observados. En general, los métodos numéricos más usados en la literatura, intentan encontrar aproximaciones que minimicen la diferencia entre las soluciones estimadas y las reales.

La implementación de métodos convencionales como el conocido como ascenso de la colina, resulta poco eficiente en problemas multimodales o no-lineales, ya que éstos son propensos a caer en óptimos locales. Otra situación que afecta a dichos métodos es el incremento de la cantidad de variables de decisión (mayor a 1000). La combinación de la no-linealidad y la complejidad de gran escala, hace más complicado el uso de los métodos de optimización. Ejemplo de esto es el conocido problema del agente de ventas viajero (*Traveling Salesman Problem*, TSP) que consiste en encontrar la ruta más corta para que el agente viaje por  $n$  ciudades teniendo permitido visitar cada ciudad una única vez. El número de posibles combinaciones, sin conocer la distribución de ciudades, es  $n!$ . De este modo, si  $n = 100$ , el número de combinaciones  $n! \approx 9.3 \times 10^{157}$ , resulta ser una cantidad inmensa.

Los métodos de optimización empleados para resolver problemas complejos, pueden catalogarse de acuerdo a la Figura 2.3.



**Figura 2.3:** Clasificación de los algoritmos de optimización.

### 2.2.2 Métodos heurísticos y metaheurísticos

El término heurístico, se define como una estrategia de prueba y error, que produce soluciones aceptables a un problema complejo, en un tiempo razonable [127]. La complejidad del problema a tratar, hace poco efectivo, incluso imposible, el uso de la búsqueda exhaustiva de todas las posibles soluciones. Uno de los retos, es encontrar los valores óptimos en un tiempo relativamente aceptable; sin embargo, no hay garantía de que las mejores soluciones puedan ser encontradas. De este modo, la idea consiste en encontrar un algoritmo eficiente y al mismo tiempo práctico, que funcione para la mayoría de los casos y sea capaz de encontrar soluciones de buena calidad. Se espera que entre las soluciones encontradas, algunas de ellas se localice cerca del óptimo global, aunque no existe garantía de que esto ocurra.

Por otra parte, la mayoría de los algoritmos metaheurísticos (AM), están inspirados en la naturaleza; en otras palabras, han sido desarrollados en base a la abstracción de algún concepto de la misma. De forma concreta, abstraer se refiere a condensar un objeto en sus partes esenciales, por lo tanto, las técnicas metaheurísticas no modelan el comportamiento fielmente, sino que solamente se basan en los principios del mismo. Se toman conceptos de la naturaleza, debido a que ella ha evolucionado desde hace millones de años y ha encontrado soluciones perfectas para casi todos los problemas.

Dos componentes principales de los AM son: a) la selección de las mejores soluciones y b) la aleatoriedad. La selección del mejor valor asegura que la solución se acercará al óptimo, mientras que la aleatoriedad evita que las soluciones sean atrapadas en un óptimo local, al mismo tiempo que incrementa la diversidad. Una buena combinación de estos dos componentes, usualmente garantiza que el óptimo global será alcanzado.

Los AM pueden ser clasificados de distintas formas, la más común de ellas, es clasificarlos atendiendo a criterios basados en población y basados en trayectoria (Figura 2.3). Dentro de los basados en población, se encuentra la inteligencia en enjambre, que se fundamenta en la interacción que poseen los individuos o agentes especializados y la manera en que de forma colectiva encuentran soluciones a los problemas complejos. Algunos ejemplos de estos comportamientos son: las colonias de hormigas, los cardúmenes de peces, colonias de abejas, manadas de mamíferos, etc.

### 2.2.3 Algoritmos metaheurísticos (AM)

La importancia que los algoritmos de optimización metaheurísticos han alcanzado en los últimos años, hace que su estudio sea atractivo para una gran cantidad de científicos en todo el mundo. Su éxito radica en la forma en que los agentes resuelven los problemas, inspirados por algún proceso de la naturaleza. En este apartado se analiza la historia de los AM y los conceptos principales que tienen en común.

#### 2.2.3.1 Historia

Uno de los avances más significativos dentro del área de los AM, es la creación de los algoritmos evolutivos (AE) en las décadas de 1960 y 1970 [127 y 128]. Dichos métodos están basados en la abstracción de las teorías de Darwin, que son representadas por los operadores de cruce o recombinación, mutación, afinidad y selectividad. Durante las décadas de 1980 y 1990 se realizaron una gran cantidad de investigaciones referentes a los algoritmos metaheurísticos [127]. Durante este periodo, se han desarrollado métodos como *Simulated Annealing* (SA) [129], propuesto por Kirkpatrick y sus colaboradores. Este algoritmo se basa en el proceso de templado de metales, que se realiza en la industria metalúrgica. Seguido de esto, Glover introdujo la técnica de *Tabu Search* (TS) la cual se considera que fue la primera en hacer uso de memorias durante la evolución del algoritmo [130].

A principios de la década de 1990, Koza propuso el primer método de evolución de programas, llamado *Genetic Programming* [131], cuya estructura es similar al lenguaje *Lisp*. De forma similar, Dorigo en su tesis doctoral, presentó el algoritmo *Ant Colony Optimization* (ACO), que trabaja con grafos completamente conectados que simulan el rastro de feromonas usado por las hormigas para comunicarse. Un análisis de la cronología de algunos importantes algoritmos que hasta la fecha han sido creados, se presenta en la

Tabla 2.1.

Nombre del algoritmo	Proceso en el que se inspira	Operadores destacados	Autores	Año de publicación
<i>Genetic Algorithm</i>	Teoría neodarwiniana de la evolución	Herencia, mutación, selección y cruzamiento	Holland	1962
<i>Evolutionary Programming</i>	Evolución	Mutación y selección	Fogel et, al.	1966
<i>Evolution Strategies</i>	Evolución	Mutación, selección, cruzamiento multipadre	Rechenberg	1971
<i>Simulated Annealing</i>	Termodinámica	Calentamiento y enfriamiento	Kirkpatrick et, al.	1983
<i>Artificial Immune Systems</i>	Sistemas inmunes biológicos	Mutación, clonación y selección	Farmer et, al.	1986
<i>Tabu Search</i>	Tabúes culturales	Condiciones tabú, aspiraciones	Glover, Laguna	1989
<i>Genetic Programming</i>	Evolución	Mutación, selección y cruza de programas	Koza	1992
<i>Ant Colony Optimization</i>	Hormigas buscando alimento	Feromonas	Dorigo	1992
<i>Particle Swarm Optimization</i>	Cardúmenes de peces, parvadas de aves	Posición y velocidad	Kennedy y Eberhart	1995
<i>Differential Evolution</i>	Evolución	Mutación	Storn y Price	1996
<i>Harmony Search</i>	Músicos buscando armonía	Improvisación y ajuste de la mejor armonía	Geem et, al.	2001
<i>Bacterial Foraging Algorithm</i>	Búsqueda de alimento de las bacterias	Reproducción, eliminación, dispersión y selección	Passino	2002
<i>Electromagnetism-like Optimization</i>	Atracción y repulsión de cargas del electromagnetismo	Cargas, fuerza, movimiento	Birbil y Fang	2003
<i>Artificial Bee Colony</i>	Abejas mieleras en búsqueda de alimento	Búsqueda de fuente de alimento, explotación y abandono	Karaboga	2005
<i>Cuckoo Search</i>	Estrategia de reproducción del pájaro cuckoo	Selección, abandono y búsqueda de nuevos nidos	Yang y Deb	2009
<i>Gravitational Search Algorithm</i>	Ley de gravitación de Newton	Fuerza, aceleración, velocidad	Rashedi et, al.	2009

Tabla 2.1: Taxonomía de los algoritmos metaheurísticos.

### 2.2.3.2 Conceptos comunes

Los algoritmos de optimización basados en procesos naturales, comparten ciertas características que hacen que sea posible su clasificación dentro de este grupo, tales características son:

1. **Inicialización:** se generan las soluciones candidatas (SC).
2. **Evaluación:** se determina la calidad de cada SC evaluándola en la función objetivo.
3. **Modificación:** se realiza una copia de cada SC, la cual se modifica mediante un cierto nivel de aleatoriedad, finalmente se evalúa para determinar su calidad.
4. **Selección:** de acuerdo a un criterio establecido, se eligen los mejores elementos existentes entre las SC y sus copias, tales valores sustituyen a los existentes en las SC.

En general, los pasos 2 a 4 se repiten hasta que se alcanza un criterio de parada, éste puede ser un número máximo de iteraciones, un valor de la función objetivo alcanzado o una cantidad de repeticiones del mismo. Tras analizar diversos métodos, es posible determinar que el paso de inicialización es básicamente el mismo para cada uno de ellos. Este paso consiste en generar valores aleatorios uniformemente distribuidos dentro de los límites del espacio de búsqueda según se indica en la ecuación (2.3).

$$x_i^d = lb_d + r (ub_d - lb_d) \quad (2.3)$$

En la ecuación (2.3),  $x_i^d$  representa el  $i$ -ésimo elemento del conjunto de SC,  $lb_d$  y  $ub_d$  son los límites inferior y superior respectivamente, para el espacio de búsqueda en la dimensión  $d$ . Por último,  $r$  es un número aleatorio uniformemente distribuido.

Por otra parte, el proceso de modificación de cada SC, resulta ser uno de los procesos clave porque es donde surte efecto el proceso en el que está inspirado el algoritmo. Dicho en otras palabras, todos los AM comparten este procedimiento, conforme al objetivo a alcanzar, pero no en cuanto a la forma de llegar al mismo. En la gran mayoría de los AM la modificación toma como punto de partida el método de Nelder y Mead [132].

## 2.3 Fundamentos de *Opposition-based learning* (OBL)

El aprendizaje basado en oposición (*Opposition-based learning*) está inspirado en la observación de lo opuesto a lo que nos rodea en todo lo que hacemos. En los últimos 2500 años el estudio de este tema ha atraído la atención de numerosos expertos [133]. Diversos algoritmos de aprendizaje automático, como los AM, han sido inspirados por

diferentes sistemas naturales, considerando la búsqueda de soluciones como una aproximación. En la mayoría de los casos, los puntos iniciales son elegidos aleatoriamente, si estos puntos son cercanos al óptimo global convergerán más rápido. Si se encuentran alejados de la solución, es decir, en la posición opuesta, la convergencia tardará más o incluso puede no converger.

### 2.3.1 Idea principal de OBL

Los elementos del concepto de lo opuesto pueden observarse en diversas áreas en nuestro entorno [134]. Este concepto es conocido mediante diferentes nombres, algunos de los que se encuentran en la literatura son entre otros los siguientes: partículas opuestas en física, complemento absoluto o relativo de un evento en la teoría de conjuntos, variables antitéticas en simulación, antónimos en lenguas, proverbios opuestos en cultura, partidos políticos de oposición, sujeto y objeto en filosofía de la ciencia o, tesis y antítesis en dialéctica.

Además, debido a la presencia de lo opuesto en el mundo real, sin importar la intensidad o la forma en que pueda influir, la naturaleza de ciertas entidades y sus contrapartes pueden ser entendidas de diferentes formas [135]. Un gran conjunto de palabras también ha sido creado para describir la diversidad y complejidad del concepto de lo opuesto: antípoda, antitético, contradictorio, contrario, dimétrico, polar, adverso, negativo, hostil, antagónico, antipático, contrario, converso, inverso, reverso, disimilar y divergente [133]. Todas estas palabras, describen alguna noción de la oposición y pueden ser empleadas en diferentes conceptos prácticos, para representar diferentes relaciones. Por lo tanto, sería complicado o imposible describir diferentes fenómenos o entidades, sin la ayuda del concepto de lo opuesto. Un ejemplo de esto, es cuando se trata de explicar los puntos cardinales, sería muy complejo definir el norte sin conocer el sur [134].

El concepto básico de *Opposition-based learning*, fue originalmente propuesto por Tizhoosh [136]. La idea principal de este concepto de optimización es encontrar una mejor solución candidata, considerando simultáneamente el valor estimado y su opuesto, de los cuales uno de los dos está más cercano al óptimo global. Desde hace algunos años este tipo de estrategia ha sido implementada en diferentes áreas del *soft computing*. Estos eficientes algoritmos metaheurísticos principalmente incluyen: *differential evolution* (DE), *particle swarm optimizacio* (PSO), aprendizaje reforzado, redes neuronales artificiales, *harmony search* (HSA), *ant colony optimization* (ACO) y *artifitial bee colony* (ABC). Una gran

cantidad de trabajos existentes en la literatura se han orientado a investigar el uso e implementaciones de OBL [137 - 139], lo cual da idea del gran campo de acción de esta técnica de aprendizaje automático.

### 2.3.2 *Conceptos básicos de OBL*

De forma general, los algoritmos de inteligencia computacional, trabajan con una población inicial e iterativamente se tratan de remplazar las soluciones actuales por alguna mejor. En ausencia de información sobre el óptimo global, se inicia con suposiciones (soluciones) aleatorias, distribuidas uniformemente en el espacio de búsqueda, ecuación (2.3). Muchas variables de los algoritmos tales como: el tiempo computacional, el uso de memoria y la complejidad de almacenamiento, tienen relación con la distancia entre las soluciones iniciales y el óptimo global. Si de forma simultánea se prueba una solución y su opuesta, el más cercano al óptimo puede ser seleccionado como solución inicial. De acuerdo con la teoría de la probabilidad, el 50% de las veces una solución candidata está más lejos que su opuesto. Por lo tanto, iniciar con el más cercano de los valores, posee un mayor potencial de acelerar la convergencia y mejorar la precisión de la optimización. Este mismo enfoque puede ser aplicado no sólo en la inicialización, sino también durante el proceso de evolución del algoritmo [140].

## 2.4 **Filtrado digital de señales**

El procesamiento digital de señales (PDS) es un área de la ciencia y la ingeniería que se ha desarrollado rápidamente durante los últimos 40 años [141]. Los ordenadores digitales y el hardware existente hace cuatro décadas eran relativamente lentos y caros. Como consecuencia, su uso se limitaba a aplicaciones de propósito general en tiempo no real, tanto científicas como comerciales. El desarrollo de la tecnología, ha permitido que el PDS, realice operaciones programables. Por medio del software se pueden modificar fácilmente las funciones del PDS para que sean ejecutadas por el hardware. Por lo tanto, el hardware digital y el software, ambos asociados, proporcionan un mayor grado de flexibilidad al diseño de sistemas.

### 2.4.1 *Conceptos básicos de procesamiento digital de señales*

Una señal se define como una cantidad física que varía con el tiempo, espacio o cualquier otra variable o variables dependientes. Toda señal contiene información que se

desea extraer o modificar de acuerdo a los requisitos de cada aplicación particular. Los sismógrafos, por ejemplo, registran señales sísmicas que contienen información sobre intensidad y características espectrales de los sismos, con ayuda de las cuales pueden determinarse entre otras cosas la ubicación de epicentros y la naturaleza de los sismos. Las señales electrocardiográficas permiten al médico determinar el estado del corazón de sus pacientes. Las señales son representadas por funciones matemáticas de una o más variables. Por ejemplo en el ámbito de los sistemas basados en reconocimiento de voz, una señal de voz se puede representar como una función dependiente de una variable temporal  $f(t)$ , las imágenes son consideradas funciones de dos variables espaciales  $f(x, y)$  y el vídeo como una señal espacio-temporal  $f(x, y, t)$ . La Tabla 2.2 presenta un resumen de las características principales empleadas para clasificar señales.

Por otra parte, un sistema se puede definir como un dispositivo que realiza una operación sobre una señal. Ejemplo de esto son los filtros que se usan para reducir el ruido y las interferencias que corrompen la señal conteniendo la información deseada. En este caso, el filtro realiza algunas operaciones sobre la señal, cuyo efecto es reducir o filtrar el ruido y la interferencia de dicha señal.

Característica	Valores	
Numero de variables	Mono-variable	Multi-variable
Dimensionalidad	Escalar (mono-canal)	Vectorial (multi-canal)
Variables independientes	Discretas	Continuas
Valores de la señal	Discretos	Continuos
Naturaleza estadística	Deterministas	Aleatorias

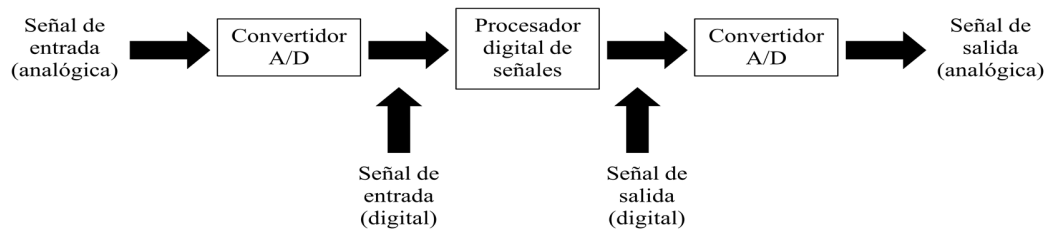
**Tabla 2.2:** Características de las señales.

Cuando una señal pasa a través de un sistema, como en el caso del filtrado, se dice que la señal ha sido procesada. En este caso, el procesamiento de dicha señal implica la separación del ruido y la interferencia. En general, el sistema se caracteriza por el tipo de operación que realiza sobre la señal. Por ejemplo, si la operación es lineal, el sistema se denomina lineal, si la operación es no lineal, el sistema se dice no lineal, etc. Tales operaciones se denominan habitualmente como procesamiento de señal.



La mayor parte de las señales que aparecen en los ámbitos de la ciencia e ingeniería son de naturaleza analógica, es decir, las señales son funciones de una variable continua, como el tiempo o el espacio y normalmente toman valores en un rango continuo. Tales señales pueden ser procesadas directamente por sistemas analógicos adecuados, tales como filtros, analizadores o multiplicadores de frecuencia, con el propósito de cambiar sus características o extraer la información deseada.

El PDS proporciona un método alternativo para procesar una señal analógica, Figura 2.4. Para realizar el procesamiento digitalmente, se necesita una interfaz entre la señal analógica y el procesador digital. Este componente es conocido como convertidor analógico-digital (A/D). En aplicaciones donde la salida del sistema digital debe entregarse de forma analógica, como en comunicaciones digitales, se debe integrar una interfaz que haga la conversión de digital a analógico (D/A).



**Figura 2.4:** Diagrama de flujo para procesar una señal analógica.

### 2.4.2 Filtros FIR

Como se mencionó en el Capítulo 1, los filtros de respuesta finita al impulso (FIR) son aquellos en los que la salida se calcula en base al valor actual y los valores previos de la entrada. En este caso, si no existe una señal de entrada, la salida va a desaparecer. Un filtro FIR se describe matemáticamente por una ecuación en diferencias de la siguiente forma:

$$y(n) = b_0x(n) + b_1x(n-1) + \cdots + b_Mx(n-M+1) = \sum_{k=0}^{M-1} b_kx(n-k) \quad (2.4)$$

donde  $M$  es la longitud del filtro,  $x(n)$  y  $y(n)$  son las señales de entrada y salida respectivamente,  $b_k$  corresponde al conjunto de coeficientes del filtro. De forma alternativa es posible expresar la salida como la convolución de la respuesta del sistema  $h(n)$  con la señal de entrada, ecuación (2.5).

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \quad (2.5)$$

Los límites del sumatorio en la ecuación (2.5), reflejan la causalidad y la duración finita del filtro. Según las ecuaciones (2.4) y (2.5) se puede deducir que  $b_k = h(k)$ . También es posible caracterizar el filtro FIR por medio de una función de transferencia, tal como se expresa en la ecuación (2.6). Esta función se puede ver como un polinomio de grado  $M - 1$  en la variable  $z^{-1}$ .

$$H(z) = \sum_{k=0}^{M-1} h(k)z^{-k} \quad (2.6)$$

### 2.4.3 Filtros IIR

En el Capítulo 1 se introdujo el concepto de los filtros de respuesta infinita al impulso (IIR), en los cuales, la salida se determina no sólo por el valor presente y previo de la entrada, sino también por los valores pasados de su propia salida. En la mayoría de las referencias bibliográficas existentes en la literatura relacionada, la representación matemática para este tipo de filtros está dada por el modelo de los filtros analógicos, de acuerdo a la siguiente ecuación:

$$H_a(s) = \frac{B(s)}{A(s)} = \frac{\sum_{k=0}^M \beta_k s^k}{\sum_{k=0}^N \alpha_k s^k} \quad (2.7)$$

donde  $\alpha_k$  y  $\beta_k$  son los coeficientes del filtro,  $B(s)$  y  $A(s)$  son la salida y la entrada del filtro respectivamente,  $M$  y  $N$  son el tamaño del filtro. Existen otras representaciones de los filtros IIR, sin embargo para fines prácticos se emplea la representación en tiempo discreto de acuerdo a la siguiente función de transferencia:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} \quad (2.8)$$

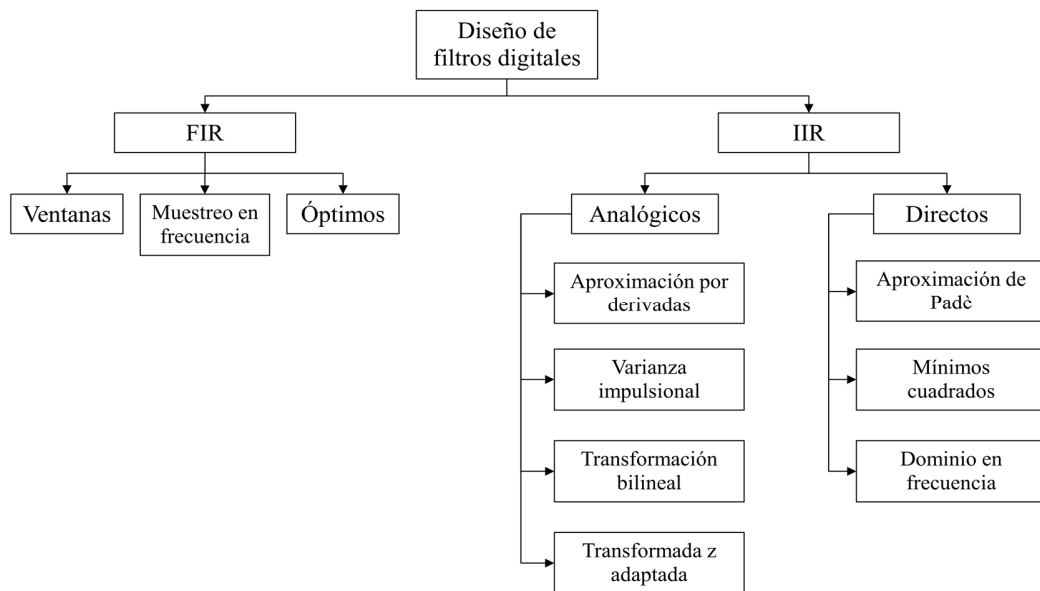
### 2.4.4 Diseño de filtros digitales

Por convención, cuando se emplea el término “filtro digital” implícitamente se hace referencia a los sistemas lineales invariantes en el tiempo (LTI). Cuando se trata de sistemas

no lineales o variantes en el tiempo se expresa tal condición. De forma general, el diseño de filtros se compone en cuatro fases:

1. **Especificación:** en esta etapa se evalúa la aplicación concreta y se construye una lista de los requisitos que debe cumplir el filtro, factores de ganancia, frecuencias de corte, bandas pasantes, rizado en las bandas pasantes y de rechazo, etc.
2. **Dimensionado:** de acuerdo a las especificaciones se dimensiona el filtro, seleccionando el orden y tipo de filtro, la estrategia de diseño, se decide si se utilizará un FIR o un IIR.
3. **Realización:** se establece aquí la estructura a utilizar, en cascada, en paralelo, así como si la aplicación permite el uso de algoritmos de optimización.
4. **Implementación:** se ocupa de aspectos de lenguaje de programación, plataforma para la realización de prototipos, sistemas integrados de desarrollo, sistemas embebidos, etc.

En general, el diseño de filtros digitales se concentra en crear un sistema que cumpla con ciertos requisitos para obtener una señal de salida con una magnitud deseada. La fase de la señal queda determinada por las consideraciones de estabilidad especialmente para los filtros IIR, mientras que para los FIR se considera la fase lineal. Existen gran variedad de técnicas para el diseño de filtros, algunas de ellas se muestran en la figura 2.5 y se ilustran con más detalle en [122].



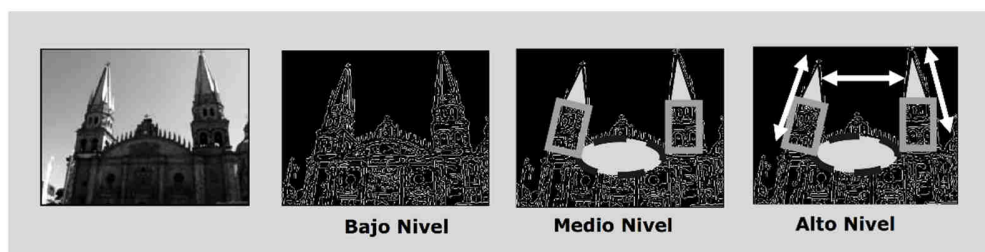
**Figura 2.5:** Taxonomía de los principales métodos para el diseño de filtros digitales.

## 2.5 Procesamiento de imágenes digitales

La visión humana puede ser considerada como un proceso que permite a una persona obtener gran cantidad de información del ambiente, la cual a su vez ayuda a ésta a desenvolverse en dicho ambiente (navegación) para realizar alguna tarea específica. El hecho de que este proceso, con todos sus problemas, se intente resolver de forma automática, formula una de las áreas de investigación y desarrollo de mayor inversión en los últimos años [143]. La Visión por Computador, se define como todo intento enfocado al desarrollo de algoritmos que traten de lograr que una máquina simule hasta cierto grado el proceso de visión biológico [143-145].

### 2.5.1 Sistemas de visión y procesamiento de imágenes

Un sistema de visión y procesamiento de imagen se compone de una serie de subsistemas que operan sobre una imagen obtenida a partir de una escena con el objetivo de interpretar alguna característica notable. La Figura 2.6 muestra cómo el sistema de procesamiento de imagen se divide en una serie de subsistemas los cuales son esencialmente: procesamiento de bajo nivel, procesamiento de nivel medio y procesamiento de alto nivel.



**Figura 2.6:** Elementos que componen el sistema de procesamiento de imagen.

En el bajo nivel, se encuentran los procesos básicos de tratamiento de imágenes tales como suavizado, umbralización, eliminación de ruido o definición de bordes, entre otros. Dichos procesos suelen conocerse en la comunidad científica como pre-procesamiento. En el nivel medio se realizan tareas tales como la definición de límites y extracción de características. En el alto nivel se establecen relaciones semánticas entre los objetos para llegar a la descripción de la escena, científicamente se habla de comprensión de imágenes (*image understanding*).

### 2.5.1.1 Procesamiento digital de imágenes

El procesamiento digital de imágenes puede definirse como el conjunto de operaciones sobre las imágenes mediante el uso de técnicas computacionales, donde el tipo de operaciones a realizar depende del nivel de procesamiento que a aplicar. Una diferencia importante entre el procesamiento de imágenes y la visión propiamente dicha, corresponde a que las imágenes con las se trabaja en el primer caso no provienen únicamente de la captación de información en el espectro electromagnético visible a la que corresponde el sistema de visión biológico, ya que las imágenes pueden obtenerse a partir de cualquier rango de longitudes de onda de dicho espectro. Así, resulta bien conocida la existencia de sistemas de procesamiento de imágenes que operan sobre imágenes generadas a partir de rayos X, gamma, infrarrojos, resonancia magnética o microondas, entre otras.

La materia prima relativa tanto del procesamiento de imágenes como a la visión propiamente dicha, son las imágenes, las cuales se consideran como la representación del mundo físico que tiene información importante, la cual es captada mediante el proceso de muestreo sobre una señal analógica, generalmente por medios electrónicos o directamente a partir de los dispositivos de captura digitales, tales como CCDs o CMOS

De forma general, para poder obtener una imagen digital, se requiere un proceso que involucra la captura, muestreo, cuantificación y codificación. Una imagen puede definirse por una función bidimensional que cuantifica los niveles de intensidad procedentes de las radiaciones electromagnéticas, incluido el espectro visible que es uno de los más comunes. Una imagen se expresa normalmente como una función  $I(x, y)$ , donde el valor de intensidad se obtiene por el indexado de las coordenadas espaciales  $x$  e  $y$ . El modelo más común de representación de la imagen es por medio de una matriz, tal que:

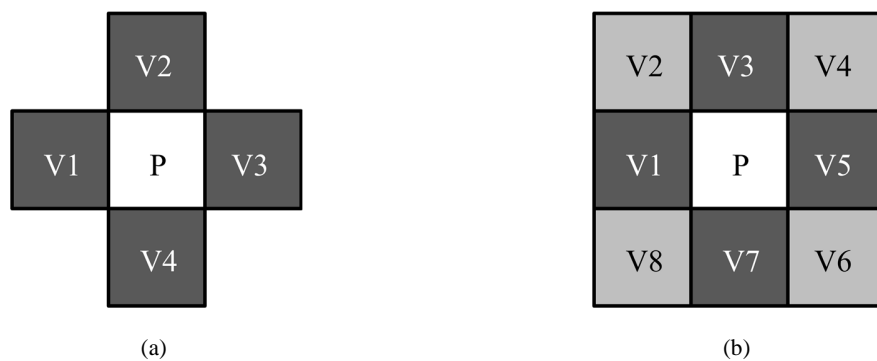
$$I(x, y) = \begin{bmatrix} I(1,1) & I(2,1) & \dots & I(N,1) \\ I(1,2) & I(2,2) & \dots & I(N,2) \\ \vdots & \vdots & \ddots & \vdots \\ I(1,M) & I(2,M) & \dots & I(N,M) \end{bmatrix} \quad (2.9)$$

### 2.5.1.2 Relaciones básicas entre píxeles

*Vecinos de un píxel:* la vecindad se define como la relación que tiene un píxel de manera posicional con los píxeles más cercanos a él. Existen dos tipos de vecindad que posee un píxel en la imagen, la vecindad de 4-vecinos y la de 8-vecinos.

La vecindad de 4-vecinos se constituye a partir de los píxeles (V1, V2, V3 y V4) que se encuentran arriba, abajo, a la derecha e izquierda del píxel en cuestión P. La Figura 2.7 (a) muestra la representación de este tipo de vecindad.

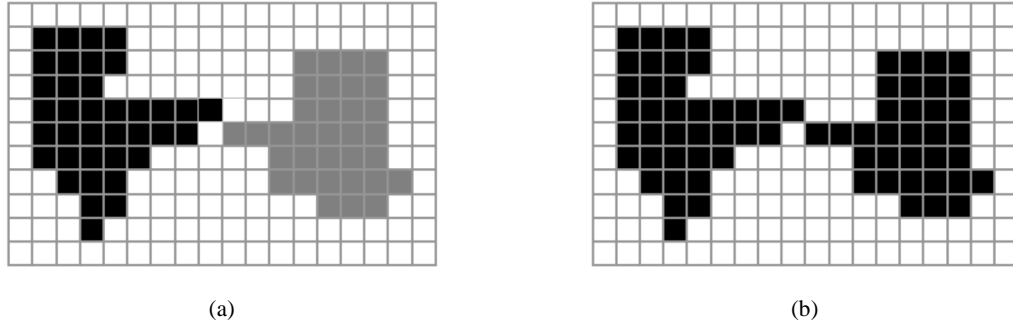
La vecindad de 8-vecinos se constituye a partir de los píxeles (V1, V2, V3, V4, V5, V6, V7 y V8) correspondientes a los 4-vecinos más, los 4 píxeles que se encuentran en forma diagonal al píxel en cuestión P. La Figura 2.7 (b) muestra la representación de este tipo de vecindad.



**Figura 2.7:** Definición de la vecindad de un píxel P, (a) Vecindad 4 y (b) vecindad 8.

*Conectividad:* la conectividad entre píxeles es un concepto utilizado ampliamente en la detección de regiones u objetos presentes en determinada imagen. Por tal motivo, la conectividad se define como una situación de adyacencia y vecindad. Bajo esta observación existen dos tipos de conectividad, la conectividad-4 y la conectividad-8.

El concepto de conectividad se puede entender mejor si se considera una imagen binaria, es decir una imagen cuyos píxeles representan una característica concreta en lugar de luminosidad, por lo que sus valores sólo pueden ser cero o uno. Considerando lo anterior se dice que dos píxeles  $I_1(x, y)$  e  $I_2(x, y)$ , cuyos valores en la imagen son uno, tienen conectividad-4 si ambos se encuentran en relación de 4-vecinos. De igual manera los mismos píxeles tendrían conectividad-8, si ambos se encuentran en relación de 8-vecinos.



**Figura 2.8:** Influencia de la vecindad en la conectividad de píxeles: (a) imagen considerando conectividad-4, (b) imagen considerando conectividad-8.

A fin de mostrar la importancia de estos conceptos, en la figura 2.8 aparece una imagen binaria en la cual se encuentran uno o dos objetos, según el tipo de conectividad utilizada. En el caso de considerar la conectividad-4, la imagen tendría dos objetos, ya que el punto donde ambas estructuras se encuentran más cercanas, los píxeles de contacto no están en relación a la vecindad 4-vecinos, Figura 2.8 (a). Por el contrario, si en la misma imagen se considera como criterio de vecindad el de 8-vecinos, los dos píxeles de contacto estarían conectados, por lo que ambas estructuras se consideran como un único objeto, Figura 2.8 (b).

### 2.5.1.3 Medidas de distancia

La distancia existente entre dos píxeles es una de las magnitudes más usadas en el procesamiento de imágenes, con aplicaciones que van desde la similitud hasta la medición de objetos encontrados en la escena. Existen varios tipos de medidas para encontrar relaciones posicionales entre los píxeles, sin embargo, las más comunes son la distancia euclidiana, la distancia *city-block* y la *chessboard*.

Para identificar y clasificar las distancias entre dos puntos se considera como referencia la imagen mostrada en la Figura 2.9(a), que contiene dos píxeles etiquetados como  $I_1(x, y) = \mathbf{p}_1$  e  $I_2(x, y) = \mathbf{p}_2$ . La distancia euclidiana se define que existe entre tales píxeles se define de acuerdo a la ecuación (2.10) y representa el vector resultante entre ellos.

$$D_E(\mathbf{p}_1, \mathbf{p}_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.10)$$

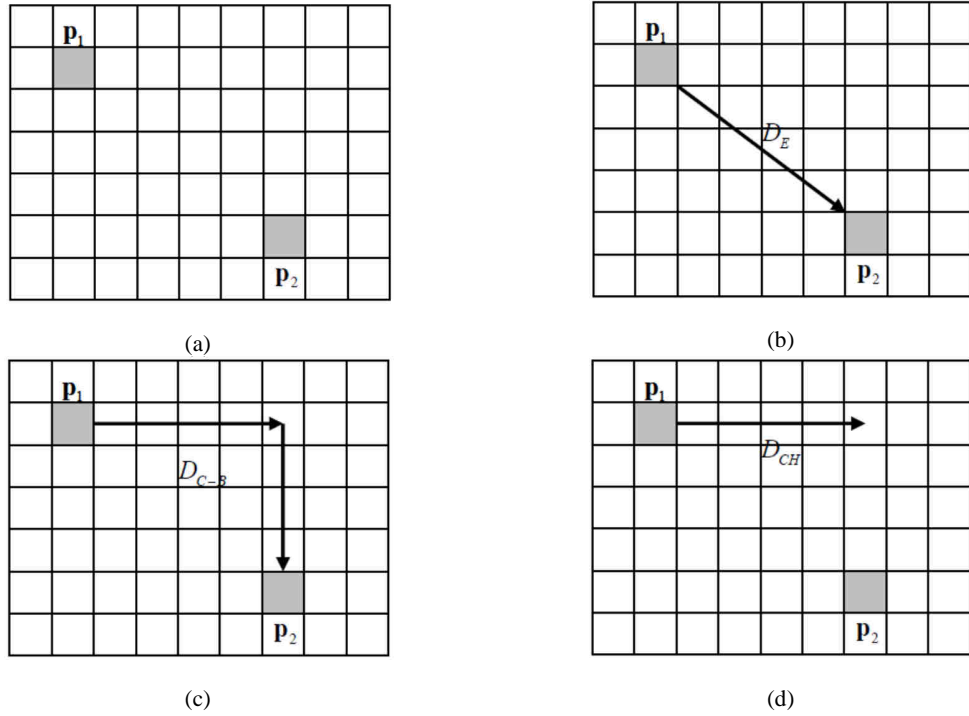
Por su parte la distancia, *city-block* corresponde a la suma de la distancia horizontal y vertical entre ambos píxeles y está definida como:

$$D_{C-B}(\mathbf{p}_1, \mathbf{p}_2) = |x_1 - x_2| + |y_1 - y_2| \quad (2.11)$$

Esta es la distancia que arroja un valor mayor en comparación a sus contrapartes euclidiana y *chessboard*, para una misma relación posicional. La distancia *chessboard* es la máxima distancia entre el recorrido horizontal y el vertical que se experimenta entre dos píxeles. Se define de acuerdo a la siguiente ecuación:

$$D_{CH}(\mathbf{p}_1, \mathbf{p}_2) = \max(|x_1 - x_2|, |y_1 - y_2|) \quad (2.12)$$

El hecho de que esta distancia sólo considere el máximo recorrido en un único sentido hace similitud al juego de ajedrez. La decisión de usar alguna de estas tres distancias depende de la implementación a realizar, sin embargo dentro de ellas, la más utilizada es con diferencia la distancia euclidiana. En la Figura 2.9 se presenta gráficamente el concepto de medida asociado con cada una de ellas.



**Figura 2.9:** Tipos de distancia para medir las relaciones posicionales entre píxeles.: (a) definición de los píxeles; (b) distancia euclidiana; (c) distancia *city-block*; (d) distancia *chessboard*.

### 2.5.2 Histogramas

Los histogramas pueden ser considerados como medidas estadísticas de la imagen y normalmente se utilizan como medio de ayuda para evaluar propiedades importantes de la imagen. Específicamente, se pueden reconocer fácilmente los errores producidos en la



captura de la imagen mediante la utilización del histograma. Aparte de poder tratar con este tipo de problemas, es posible también procesar la imagen tomando como base el histograma, el cual puede aplicarse para mejorar la imagen o bien para resaltar características de ella, que pueden ser solamente extraídas o analizadas en fases de procesamiento posteriores, por ejemplo, en una fase de reconocimiento de patrones.

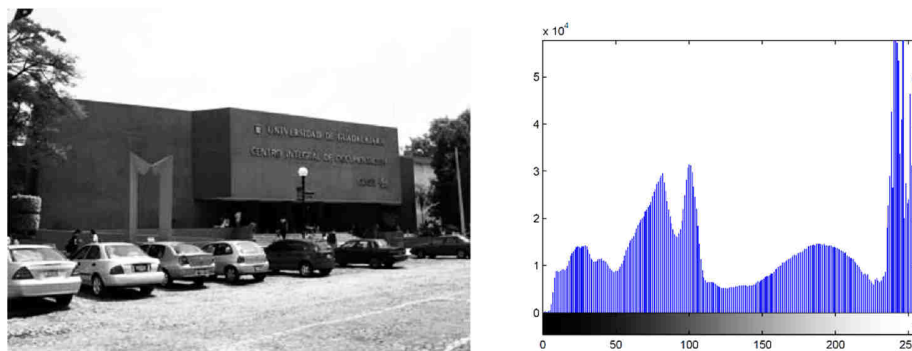
### 2.5.2.1 ¿Qué es un histograma?

Los histogramas son distribuciones que describen la frecuencia con la que se presentan los valores de intensidad de los píxeles en la imagen. En el caso más sencillo, los histogramas se obtienen a partir de imágenes en escala de grises, un ejemplo es el mostrado en la Figura 2.10. Para una imagen en escala de grises  $I(u, v)$ , con valores de intensidad en el intervalo  $[0, K-1]$ , tiene un histograma  $H$  que contendrá exactamente  $K$  diferentes valores. Considerando una típica imagen en escala de grises de 8 bits será  $H = 2^8 = 256$ . Cada valor del histograma se define como  $h(i)$  y es igual al número de píxeles de  $I$  con el valor de intensidad  $i$  para todos los valores  $0 \leq i < K$ . Formalmente se expresa como:

$$h(i) = \text{card} \{ (u, v) | I(u, v) = i \} \quad (2.13)$$

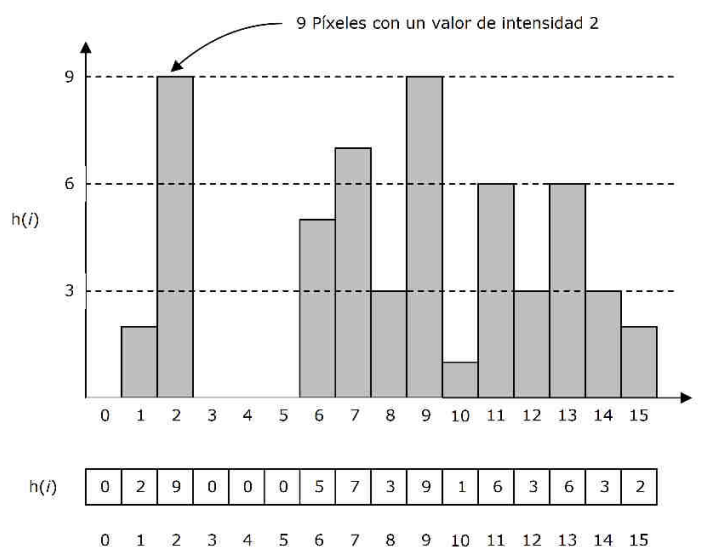
$h(0)$  es entonces el número de píxeles con el valor 0,  $h(1)$  el número de píxeles que tienen el valor de 1, así sucesivamente mientras que finalmente  $h(255)$  representa el número de píxeles con el máximo valor de intensidad (en este caso píxeles en blanco) de la imagen, finalmente  $\text{card} \{ \dots \}$  representa la cardinalidad del número de elementos. Como resultado del cálculo del histograma se obtiene un vector unidimensional con una longitud  $K$ , tal y como se presenta en la Figura 2.11 donde en este caso concreto  $K = 16$ .

Evidentemente, el histograma no proporciona información acerca del origen de los píxeles que lo conforman. Esto es, pérdida de información de la relación espacial que tenían los píxeles en la imagen original, por lo que resulta imposible reconstruir una imagen a partir de su histograma. Para evidenciar este hecho la Figura 2.12 muestra tres diferentes imágenes que producen el mismo histograma.



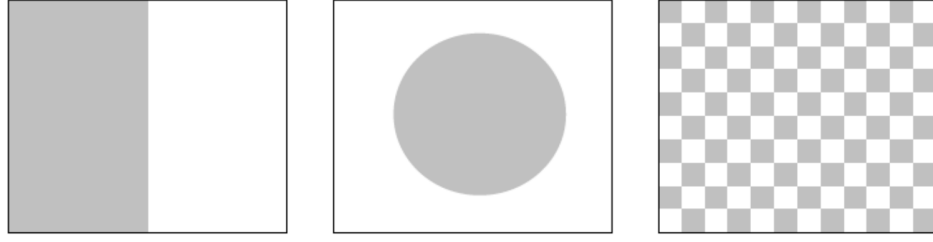
**Figura 2.10:** Imagen de intensidad en escala de grises de 8 bits con su respectivo histograma.

En la Figura 2.11 el vector de un histograma como ejemplo, con 16 posibles valores de intensidad. El índice de los elementos del vector  $i = 0 \dots 15$  representa el valor de la intensidad. El valor de 9 en el elemento 2, significa que en la correspondiente imagen, el valor de la intensidad 2 se repite 9 veces.



**Figura 2.11:** Ejemplo de representación de un histograma.

El histograma muestra importantes características de una imagen, como por ejemplo, el contraste y el rango dinámico, los cuales son problemas atribuidos a la captura de la misma. Tales problemas, deben ser resueltos para que la imagen pueda ser utilizada con mejores propiedades, en procesamiento posteriores.



**Figura 2.12:** Tres diferentes imágenes que producen el mismo histograma.

### 2.5.3 Segmentación

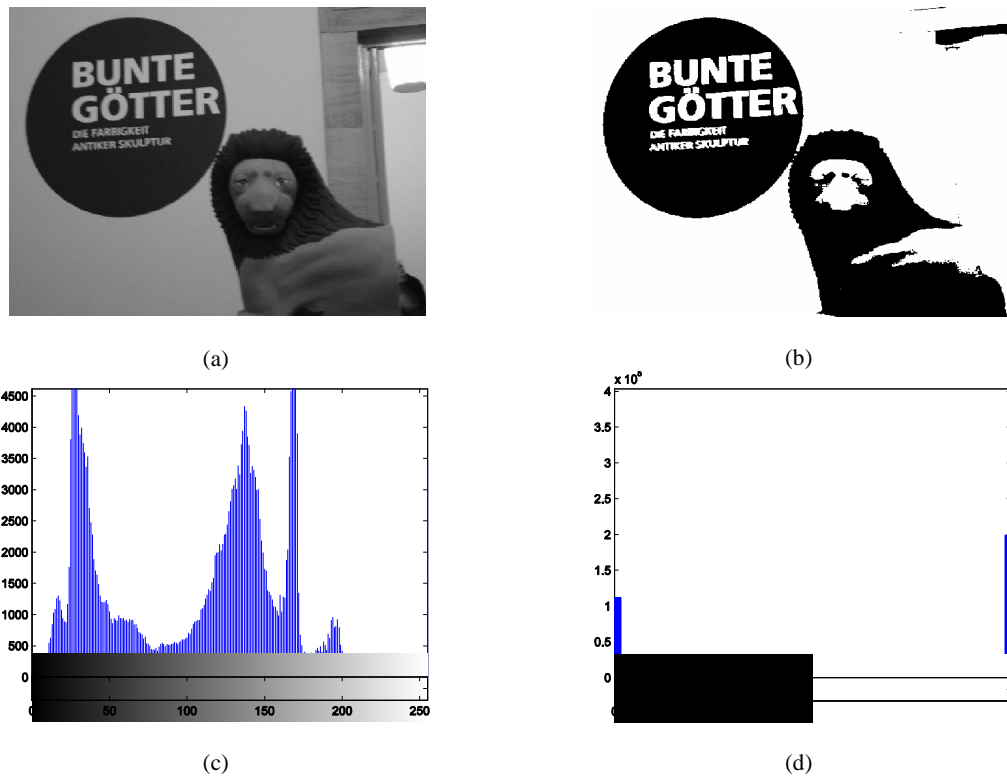
De forma general, la operación de segmentación se define como un proceso que divide una imagen en sus regiones más representativas, esto es posible, considerando que al menos existe un criterio que resalta la importancia de dichas regiones. Tomando la definición de segmentación explicada por Gonzalez y Woods [145], tenemos:

Sea  $f(x, y)$  una imagen y  $R$  la representación completa de la imagen. De acuerdo a la segmentación, es posible decir que  $R$  contiene  $n$  subregiones  $R_i$ , donde  $i = 1, 2, \dots, n$ , de tal forma que se cumpla lo siguiente:

- a)  $\bigcup_{i=1}^n R_i = R$ ,
- b)  $R_i$  tiene algún tipo de conectividad (sección 2.5.1),  $i = 1, 2, \dots, n$ ,
- c)  $R_i \cap R_j = \emptyset$  para todo  $i$  y  $j$ ,  $i \neq j$ ,
- d)  $P(R_i) = \text{VERDADERO}$  para  $i = 1, 2, \dots, n$ ,
- e)  $P(R_i \cup R_j) = \text{FALSO}$  para  $i \neq j$ ,

Donde  $P(R_i)$  es un predicado lógico sobre los puntos del conjunto  $R_i$  y  $\emptyset$  es el conjunto vacío. La condición a) indica que la segmentación debe ser completa; esto es que cada píxel debe estar contenido dentro de alguna región. La siguiente condición se refiere a que los puntos de una región deben tener algún tipo de conectividad. La condición c) indica que las regiones deben ser disjuntas. La condición d) trata de las propiedades que deben satisfacer los píxeles de una región segmentada, por ejemplo  $P(R_i) = \text{VERDADERO}$  si todos los píxeles de  $R_i$  tienen la misma intensidad. Por último, la condición e) indica que las regiones  $R_i$  y  $R_j$  son diferentes en el sentido del predicado  $P$ .

La segmentación se puede considerar como uno de los principales problemas que debe tratarse en un sistema de visión. Un buen rendimiento de este proceso se ve reflejado en el desempeño del sistema total, ya que otros procesos dependen por completo de sus resultados. Existen básicamente dos criterios en los que están basados los algoritmos de segmentación: la homogeneidad de una región (mejor conocido como umbralado o umbralización) y la discontinuidad entre las regiones adyacentes. Independientemente del tipo de segmentación que se realice, su aplicación en imágenes nos permite su partición, con esto es posible analizar y procesar sólo las regiones que sean de interés. Este procesamiento basado en particiones, ofrece características tales como: reducción en el uso de memoria, incremento en la velocidad de procesamiento e incremento en la precisión del sistema.



**Figura 2.13:** Ejemplo de aplicación de la segmentación bi-nivel sobre una imagen, considerando  $th = 80$ ,  $p_0 = 0$  y  $p_1 = 1$ ; (a) imagen en escala de grises; (b) imagen binarizada; (c) y (d) sus respectivos histogramas.

Los métodos empleados para la segmentación se clasifican en base a la cantidad de umbrales que se deben aplicar, las categorías en que se dividen son: bi-nivel y multinivel. Para la segmentación bi-nivel se selecciona un único valor de umbral, el cual divide los píxeles de la imagen en dos clases, la primera representa los objetos y la segunda el fondo

de la imagen. En la segmentación multinivel, se requieren más de dos valores de umbral, esto permite una segmentación más adecuada de los diversos objetos de la imagen. De acuerdo a la información de la imagen las técnicas de segmentación también pueden ser catalogadas en: umbralización (o umbralado) y crecimiento de regiones, las cuales se tratarán en las siguientes subsecciones.

### 2.5.3.1 Segmentación por umbralización

Debido a su sencillez, la umbralización es una de las técnicas más usadas para segmentar imágenes digitales. En su implementación más básica la idea consiste en convertir una imagen en escala de grises  $f(x, y)$  en una imagen binaria  $b(x, y)$ , el objetivo de este proceso es obtener una separación entre los objetos de interés y el fondo de la imagen. Con esto considera que las intensidades de los píxeles de una imagen deben dividirse en dos clases, dichas clases dependen de un valor de umbral (*threshold*) predefinido  $th$ . Los píxeles ( $p$ ) de la imagen asumirán dos posibles valores  $p_0$  o  $p_1$  dependiendo de su valor de intensidad, la transformación de la imagen original en binaria por umbralado se define como:

$$b(x, y) = \begin{cases} p_0 & \text{si } f(x, y) < th \\ p_1 & \text{si } f(x, y) \geq th \end{cases} \quad (2.14)$$

los valores de  $x$  e  $y$  corresponden a la posición del píxel en cuestión, por otra parte, los valores que toman los píxeles en la imagen binaria, comúnmente son considerados los valores mínimo y máximo del histograma, es decir,  $p_0 = 0$  y  $p_1 = L - 1$  considerando  $L = 256$ , desde un punto de vista de la lógica algebraica  $p_0 = 0$  y  $p_1 = 1$ . Este procedimiento se conoce como binarización o segmentación bi-nivel. Un ejemplo de aplicar este tipo de umbralización se presenta en la Figura 2.13. El efecto de la binarización se observa claramente en el histograma de la imagen, donde la distribución está claramente dividida en dos partes, considerando como la frontera de división el valor de umbral.

La definición de umbralización se puede extender a una segmentación multinivel, en tal caso se requiere un conjunto de valores de umbral  $\mathbf{TH} = [th_1, th_2 \dots th_n]$ . De este modo, los píxeles de la imagen serán divididos en  $n + 1$  clases. Cada píxel de la imagen segmentada  $f_{th}(x, y)$  se asigna de acuerdo a la siguiente ecuación.

$$f_{th}(x, y) = \begin{cases} th_1 & \text{si } f(x, y) < th_1 \\ th_2 & \text{si } th_1 \leq f(x, y) < th_2 \\ \vdots & \vdots \\ th_n & \text{si } th_{n-1} \leq f(x, y) < th_n \\ L-1 & \text{si } th_n \leq f(x, y) < L-1 \end{cases} \quad (2.15)$$

Otra forma en que son conocidos los métodos de umbralado es como técnicas de agrupamiento estadístico de píxeles, esto es así porque dichos métodos emplean propiedades estadísticas para realizar la búsqueda del mejor valor de umbral. De acuerdo a la información que utilizan, estos métodos se pueden clasificar de la siguiente manera:

- **Forma del histograma:** donde se considera la información contenida en el histograma suavizado de la imagen.
- **Agrupamiento director de los píxeles:** este método es también conocido como agrupamiento o *clustering* y se basa en agrupar las muestras de acuerdo a diferentes niveles de intensidad.
- **Entropía:** se calcula la entropía existente en las regiones que corresponden a las clases, del mismo modo se calcula la entropía cruzada entre la imagen original y la imagen particionada.
- **Lógica difusa:** miden la similitud entre la imagen original y la umbralizada usando medidas difusas.
- **Globales:** consideran la correlación existente entre los píxeles a una escala global.
- **Locales:** adaptan los valores de umbral regionalmente, dependiendo de las características de la imagen en sus diferentes regiones.

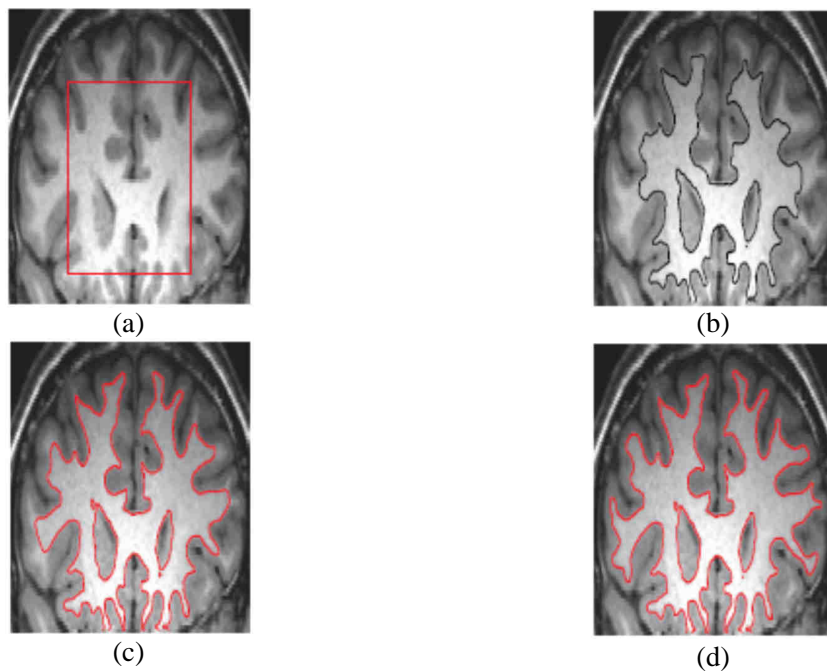
En la Tabla 2.3 se muestra una lista de métodos de umbralización [64,34 146-153] de acuerdo a [67].

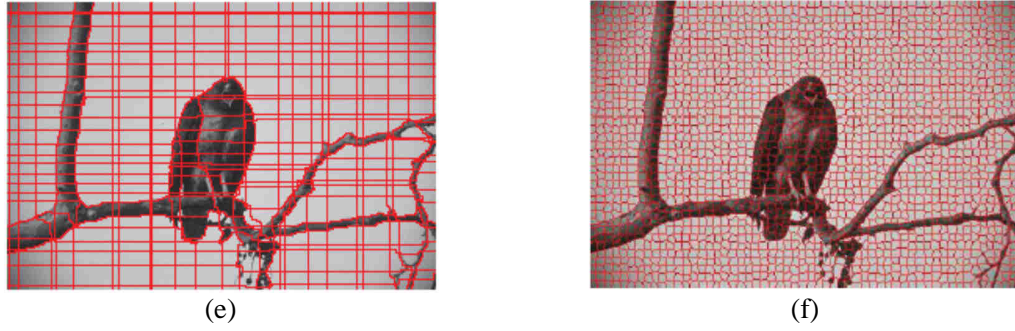
Nombre	Autor	Información utilizada
Concha convexa	Rosenfeld	Forma del histograma
Picos y valles-	Sezan	Forma del histograma
Modelado de la forma	Olivio	Forma del histograma
Modelado de la forma	Guo	Forma del histograma
Umbralado iterativo	Riddler	Agrupamiento de píxeles
Umbralado iterativo	Yanni	Agrupamiento de píxeles
Umbralado por agrupamiento	Otsu	Agrupamiento de píxeles
Umbralado por agrupamiento	Kapur	Agrupamiento de píxeles
Umbralado espacial	Beghdadi	Dependencia espacial de los píxeles
Umbralado adaptivo	Sauvola	Dependencia local de los píxeles

**Tabla 2.3:** Métodos de segmentación por búsqueda de umbrales.

### 2.5.3.2 Segmentación por crecimiento de regiones

Estos métodos se usan para extraer objetos de la imagen mediante la partición de la misma utilizando la información que proporcionan los píxeles de la imagen (varios niveles de intensidad) junto con otras características tales como bordes o líneas entre otras. Dentro de esta clase de técnicas se clasifican los algoritmos que utilizan contornos activos, mejor conocidos en terminología inglesa como *snakes* [154, 155], líneas de nivel o algoritmos que realizan segmentación sobre regiones rectangulares de la imagen original, por ejemplo superpíxeles, entre otros. Dentro de la última categoría, un superpíxel se define como la representación de un conjunto de píxeles que son localmente homogéneos [156]. Existen diversas variantes del método original que definió los superpíxeles [157], como es el caso del algoritmo de agrupamiento lineal simple (*Simple Linear Iterative Clustering Algorithm* (SLIC)) [158], el método de enmallado [159], entre otros [160,161]. La Figura 2.14 muestra los resultados de la implementación de algunos algoritmos de segmentación por crecimiento de regiones [162, 163].





**Figura 2.14:** (a), (b), (c), (d), resultados tras aplicar un algoritmo de segmentación por contornos activos. (e) y (f) son los resultados tras segmentar por superpíxeles

#### 2.5.4 Detección de líneas y curvas

La detección de rasgos geométricos en imágenes digitales es de gran ayuda para identificar objetos específicos. Para llevar esto a cabo en los sistemas de visión se cuenta con técnicas que permiten extraer características de la imagen. En el procesamiento de imágenes existen índices visuales (bordes, puntos y líneas de interés) que pueden ser usados para seguir o detectar objetos. En su mayoría los métodos para extracción de índices o características visuales, se basan en imágenes binarias, las cuales se obtienen tras la aplicación de un método de umbralización.

##### 2.5.4.1 Rasgos geométricos básicos

A continuación se describen los rasgos geométricos que pueden calcularse a partir de los píxeles de un objeto en una imagen binaria:

*Perímetro de un objeto:* se refiere a la longitud del contorno de un objeto  $O$ . Para el cálculo del perímetro, es necesario considerar el tipo de vecindad a usar, ya que la distancia del perímetro de un contorno bajo una vecindad 4-vecinos es mayor al de 8-vecinos. El perímetro entonces, puede calcularse mediante la siguiente expresión,

$$Perimetro(O) = \sum_{i=1}^M longitud(c_i) \quad (2.16)$$

donde:

$$longitud(c_i) = \begin{cases} 1 & \text{para } c = 0, 2, 4, 6 \\ \sqrt{2} & \text{para } c = 1, 3, 5, 7 \end{cases} \quad (2.17)$$

El valor del perímetro calculado por las ecuaciones (2.16) y (2.17) determina las



distancias verdaderas a lo largo del contorno. En la práctica normalmente este valor es ajustado para obtener un nuevo valor del perímetro definido como  $U(O)$ , que se calcula mediante la siguiente expresión.

$$U(O) = 0.95 * \text{Perímetro}(O) \quad (2.18)$$

*Área de un objeto:* se refiere a su tamaño. El área de un objeto  $O$  bidimensional binario, se define como el número de píxeles que conforman dicho objeto, definida como:

$$\text{Area}(O) = N = |O| \quad (2.19)$$

Cuando el objeto no se encuentra descrito por un conjunto agrupado de píxeles, sino por el contorno que lo rodea es posible aproximar su área a partir del contorno exterior cerrado, sólo si en el objeto no existen contornos interiores, mediante la aplicación de la ecuación siguiente:

$$\text{Area}(O) = \frac{1}{2} \cdot \left| \sum_{i=1}^M \left( x_i \cdot y_{[(i+1) \bmod (M)]} - x_{[(i+1) \bmod (M)]} \cdot y_i \right) \right| \quad (2.20)$$

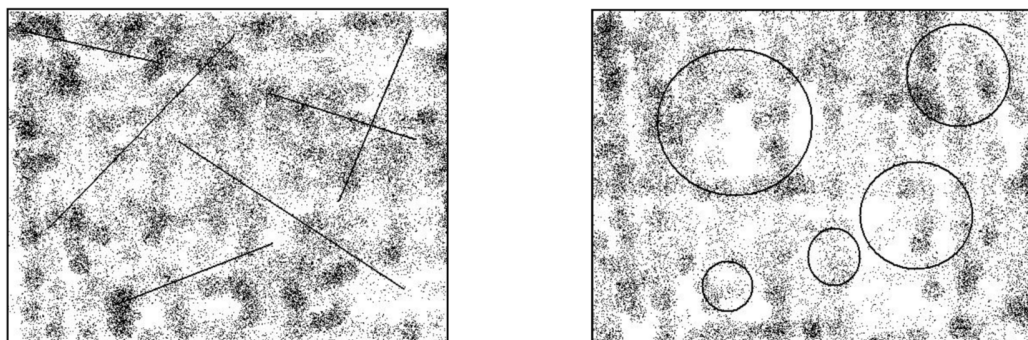
donde  $x_i$  y  $y_i$  son las coordenadas de los  $M$  puntos que conforman el contorno cerrado del objeto,  $\bmod(\bullet)$  corresponde a la operación modulo. Es importante mencionar, que tanto el perímetro como el área, son características geométricas robustas ante el desplazamiento y la rotación, sin embargo son altamente sensibles al escalamiento.

#### 2.5.4.2 Estructuras en una imagen

Un enfoque intuitivo para encontrar estructuras en una imagen podría consistir en la búsqueda de puntos de borde conectados, de forma que empezando por un determinado punto perteneciente a un borde, paso a paso añadir los píxeles que pertenecen al borde completo y con ellos determinar la estructura. La aproximación anterior puede aplicarse tanto a imágenes provenientes de una umbralización, así como a imágenes provenientes de algún tipo de segmentación. Esta aproximación sin embargo fallará, al no considerar las fracturas y ramificaciones en los bordes, producto del ruido y la incertidumbre propia de los algoritmos de cálculo del gradiente o la segmentación, los cuales no incorporan ninguna clase de criterio sobre las formas que se buscan en la imagen.

Un enfoque totalmente diferente es la búsqueda global de estructuras presentes en la imagen, que de alguna manera se aproximan o relacionan a un tipo de forma previamente especificada. Como puede verse en la Figura 2.15, a los ojos de un humano resultan

claramente diferenciables los tipos de estructuras presentes en las imágenes a pesar de la existencia de una gran cantidad de píxeles añadidos. Es hasta ahora desconocido el mecanismo biológico responsable que permite relacionar y reconocer estructuras en las imágenes percibidas por humanos o animales. Una técnica que permite, al menos desde el punto de vista computacional resolver, este problema es la llamada transformada de Hough la cual se describe en la siguiente subsección.



**Figura 2.15:** Imágenes de ejemplo en las cuales el ojo humano puede distinguir los objetos existentes a pesar de los puntos añadidos como ruido a la imagen.

#### 2.5.4.3 La transformada de Hough

El método de la transformada de Hough ideado por Paul Hough y patentado en Los Estados Unidos de América es normalmente conocido en la comunidad de visión por computadora con ese mismo nombre, esto es como la transformada de Hough [143-145, 155]. Dicha transformada permite localizar formas paramétricas a partir de una distribución de puntos presentes en una imagen. Por formas paramétricas se entienden líneas, círculos, elipses u otras estructuras que pueden ser descritas mediante la utilización de pocos parámetros que describen una ecuación característica. Debido a que este tipo de estructuras paramétricas (líneas, círculos y elipses), como muestra la Figura 2.16, se presentan con frecuencia en imágenes, es de especial importancia encontrarlas de forma automática mediante técnicas computacionales.



**Figura 2.16:** Los objetos paramétricos aparecen como líneas, círculos o elipses, aparecen con frecuencia en las imágenes digitales.

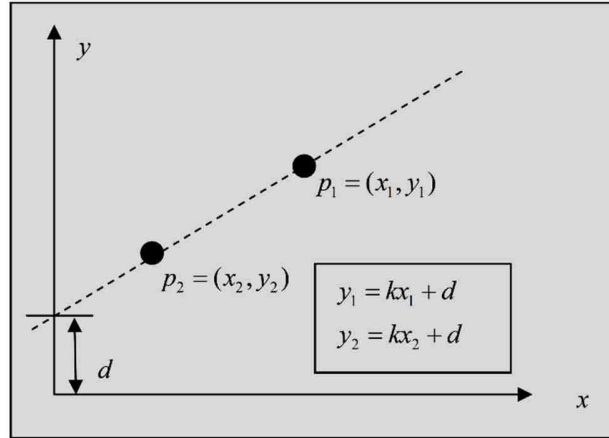
La transformada de Hough, es frecuentemente empleada en los sistemas de visión y procesamiento de imágenes para la detección de líneas en imágenes binarias producidas por umbralización. Una línea en un espacio bidimensional se define mediante el uso de los parámetros reales tal como se expresa en la siguiente ecuación:

$$y = kx + d \quad (2.21)$$

donde  $k$  representa la pendiente y  $d$  es el punto del eje  $y$  donde la línea lo intercepta, Figura 2.17. Una línea que pasa por dos diferentes puntos  $p_1 = (x_1, y_1)$  y  $p_2 = (x_2, y_2)$  debe satisfacer las siguientes condiciones:

$$y_1 = kx_1 + d \text{ y } y_2 = kx_2 + d ; k, d \in \mathbb{R} \quad (2.22)$$

Considerando la transformada de Hough, el objetivo por lo tanto consiste en estimar los parámetros  $k$  y  $d$  de una línea, que pasa por diferentes puntos pertenecientes a los bordes de un objeto. Ante esta situación, surge la pregunta: ¿cómo se puede determinar cuántos posibles puntos engloba una línea? Un método para resolver este problema consistiría trazar todas las líneas posibles y contar exactamente los puntos que pasan sobre cada una de ellas, para después borrar las que contienen sólo un determinado número de puntos. Lo anterior es posible para imágenes de un tamaño reducido, sin embargo para tamaños mayores es completamente ineficiente [143].



**Figura 2.17:** Dos puntos  $p_1$  y  $p_2$  que pertenecen a la misma línea, por lo que para determinar la ecuación de la recta habrá que estimar  $k$  y  $d$ .

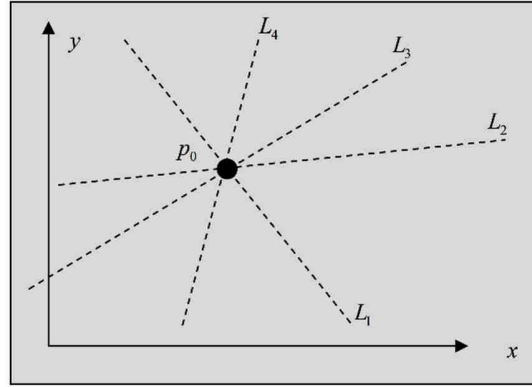
La trasformada de Hough aborda el problema de la detección de líneas de una manera ligeramente diferente, al producir todas las posibles líneas que pasan a través de un píxel el cual corresponde a un borde de la imagen. Cada línea  $L_p$  que pasa a través de un punto  $p_0 = (x_0, y_0)$  posee la siguiente ecuación:

$$L_p = y_0 = kx_0 + d \quad (2.23)$$

donde los vectores  $k$  y  $d$  cambian su valor para poder trazar todas las líneas posibles que tienen en común  $x_0$  y  $y_0$ . El conjunto de soluciones para  $k$  y  $d$  en la ecuación (2.23), corresponde a una cantidad infinita de rectas que pasan por el punto  $p_0$ , un ejemplo de esto se muestra en la Figura 2.18. Para un determinado valor de  $k$  se obtiene la correspondiente solución de la ecuación (2.23) en función de  $d$ , la cual determina la siguiente expresión.

$$d = y_0 - kx_0 \quad (2.24)$$

Aquí se representa de nuevo una función lineal, donde  $k$  y  $d$  son las variables y  $x_0$  e  $y_0$  las constantes de los parámetros considerados en la función. El conjunto de soluciones  $\{k, d\}$  de la ecuación (2.24) describe los parámetros de todas las posibles rectas  $L_p$  que pasan por  $p_0 = (x_0, y_0)$ .



**Figura 2.18:** Conjunto de rectas que pasan a través de un punto  $p_0$  en común. Todas las posibles líneas  $L_p$  que pasan por  $p_0$  tienen la expresión  $y_0 = kx_0 + d$ .

Partiendo de lo anterior, para un determinado píxel de la imagen  $p_i = (x_i, y_i)$  le corresponde un conjunto de rectas definidas como:

$$R_i : d = y_i - kx_i \quad (2.25)$$

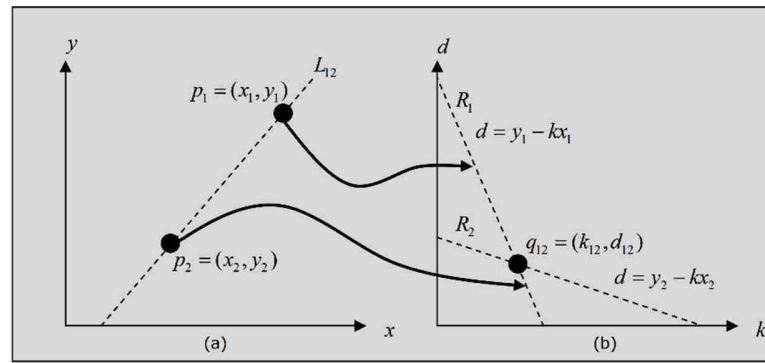
es aquí donde las variables  $k$  y  $d$  definen el espacio de parámetros de Hough, mientras que  $x_i$  y  $y_i$  son conocidos como espacio de parámetros de la imagen. La relación existente entre ambos espacios se presenta en la Tabla 2.4.

Espacio de parámetros de la imagen $(x, y)$	Espacio de parámetros de Hough $(k, d)$
Punto: $p_i = (x_i, y_i)$	Línea: $R : d = y_i - kx_i$
Línea: $L : y = k_j x + d_j$	Punto: $q_j = (k_j, d_j)$

**Tabla 2.4:** Relación existente de puntos y líneas entre los diferente espacios de parámetros.

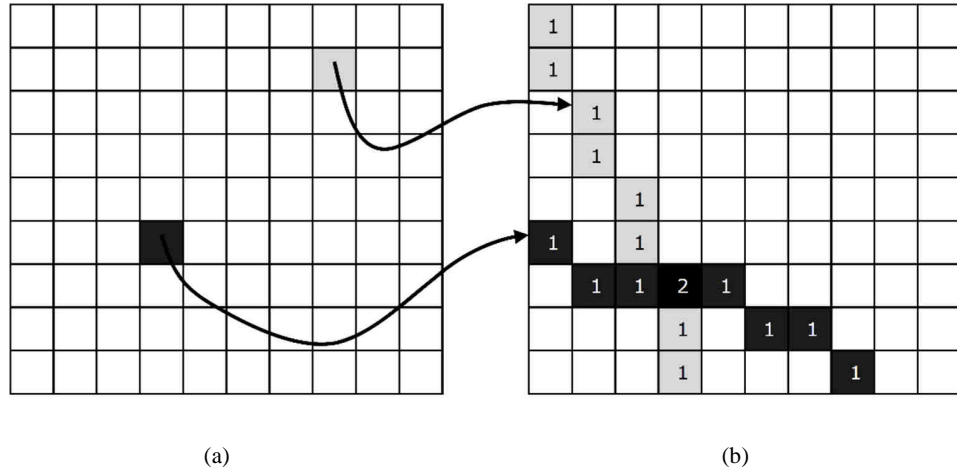
Cada punto  $p_i$  en el espacio de parámetros de la imagen corresponde a una recta en el espacio de parámetros de Hough. Considerando lo anterior, se está interesado en aquellos puntos donde las rectas establecidas en el espacio de parámetros de Hough se interceptan, ya que corresponderán a los valores de  $k$  y  $d$ , que representan a la recta del espacio de la imagen, que a su vez pasa por los puntos que definieron las rectas en el espacio de parámetros de Hough. Como se muestra en la Figura 2.19, las rectas  $R_1$  y  $R_2$  se cortan en el punto  $q = (k_{1,2}, d_{1,2})$  en el espacio de parámetros de Hough, las cuales representan a los

puntos  $p_1$  y  $p_2$  en el espacio de parámetros de la imagen. Por lo que la recta en el espacio de la imagen tendrá la pendiente de  $k_{1,2}$  y un punto de intercepción en el eje  $y$  de  $d_{1,2}$ . El número de rectas que se cortan en el espacio de parámetros de Hough indica que la línea en el espacio de la imagen estará formada por ese número de puntos. Por lo que podría establecerse que: “si hay un número de rectas  $NR$  que se interceptan en  $(\bar{k}, \bar{d})$  del espacio del parámetros de Hough, entonces habrá  $NR$  puntos que se encuentran sobre la línea  $y = \bar{k}x + \bar{d}$  en el espacio de la imagen”.



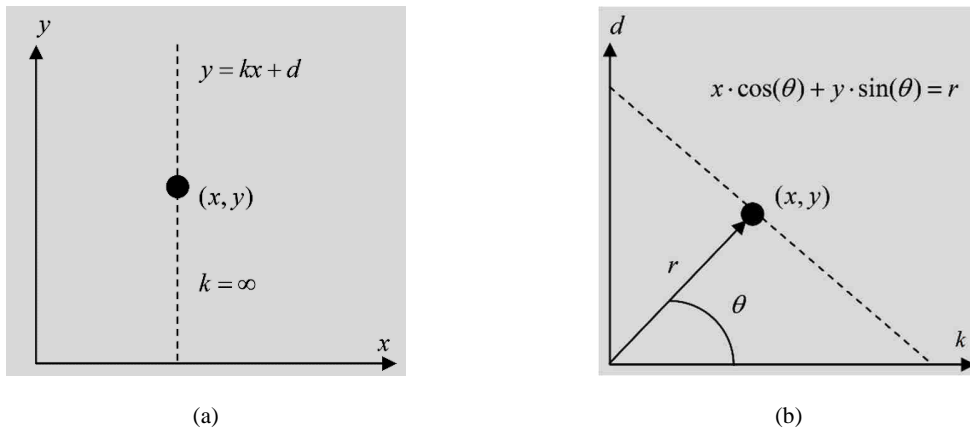
**Figura 2.19:** Relación entre el espacio de parámetros de la imagen y el espacio de parámetros de Hough; (a) espacio de parámetros de la imagen; (b) espacio de parámetros de Hough.

El método para la localización de rectas en una imagen se fundamenta en encontrar las coordenadas en el espacio de parámetros de Hough donde varias líneas se cortan. Para el cálculo de la trasformada de Hough se necesita primeramente discretizar de manera escalonada el intervalo de valores que corresponden a  $k$  y  $d$ . Al momento de realizar el conteo de cortes producidos por la intercepción de varias rectas en el espacio de parámetros de Hough, se utiliza una matriz de registros acumuladores donde cada celda es incrementada en función de la cantidad de rectas que pasan por las coordenadas que definen esa celda. De tal manera que el número final  $N_p$  almacenado en el registro significaría que esa recta en específico, se encuentra constituida de  $N_p$  píxeles del espacio de parámetros de la imagen. La Figura 2.20 ilustra este proceso considerando como ejemplo el tomado en la Figura 2.19.



**Figura 2.20:** Idea fundamental de la transformada de Hough; (a) espacio de parámetros de la imagen y (b) acumulador del espacio de parámetros de Hough. La matriz de registros de acumulación es la versión discreta del espacio de Hough.

De acuerdo a la Figura 2.20, a cada punto en la imagen le corresponde una línea en el espacio de Hough. La operación que se realiza es aditiva, lo que indica que cada celda de la matriz es incrementada en uno conforme una recta pase por él. De esta forma, los puntos con un valor máximo local en el espacio de parámetros de Hough determinan los valores  $k$  y  $d$  que representan a las rectas de la imagen.



**Figura 2.21:** Diferentes tipos de parámetros para caracterizar líneas; (a) se muestra la singularidad para el caso de una línea vertical (ecuación (2.20)); (b) presenta el uso del modelo que evita el error computacional (ecuación (2.25)).

Para obtener los resultados finales tras aplicar la transformada de Hough, falta realizar un cambio en el modelo paramétrico. Esto es debido a que se produce un error computacional ( $k = \infty$ ) en las líneas verticales si se aplica directamente la ecuación (2.21). Una mejor opción es el uso de la siguiente ecuación de la recta en coordenadas polares:

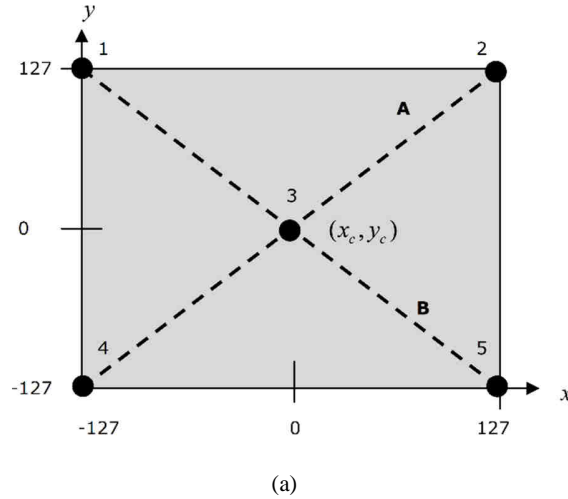
$$x \cdot \cos(\theta) + y \cdot \sin(\theta) = r \quad (2.26)$$

esta forma de representación paramétrica no presenta ninguna singularidad y además permite una cuantificación lineal de sus parámetros  $r$  y  $\theta$ . La Figura 2.21 ilustra el problema de singularidad de la ecuación (2.21) y muestra la manera de relacionar la ecuación (2.26) con la de la recta. Con la utilización de la ecuación (2.22) para la descripción de rectas, el espacio de parámetros de Hough varía. Por lo que el espacio de parámetros de las coordenadas  $r$ ,  $\theta$  y el punto  $p_i = (x_i, y_i)$  del espacio de parámetros de la imagen, se relacionan mediante la siguiente expresión:

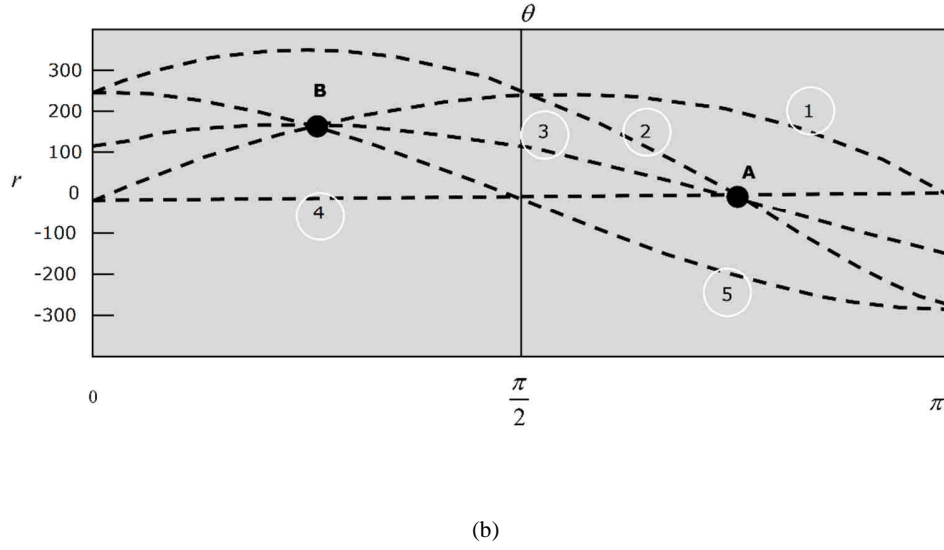
$$r_{x_i, y_i} = x_i \cdot \cos(\theta) + y_i \cdot \sin(\theta) \quad (2.27)$$

donde el intervalo de valores para  $\theta$  es  $0 \leq \theta < \pi$ , Figura 2.22. Si se utiliza el centro de la imagen como punto de referencia para definir coordenadas de los píxeles existirán píxeles negativos y positivos, entonces el intervalo de valores para  $r$  es  $-r_{\max} \leq r_{x, y} \leq r_{\max}$ , es decir se restringe a la mitad del tamaño de la imagen, suponiendo que dicho tamaño es  $M, N$ , así el valor máximo de  $r$  se define como sigue,

$$r_{\max} = \sqrt{\left(\frac{M}{2}\right)^2 + \left(\frac{N}{2}\right)^2} \quad (2.28)$$





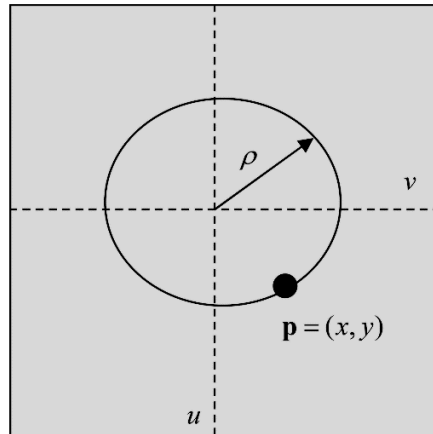


**Figura 2.22:** (a) Espacio de parámetros de la imagen y (b) espacio de parámetros de Hough usando el modelo de la recta de la ecuación (2.26).

#### 2.5.4.4 Transformada de Hough para la detección de círculos

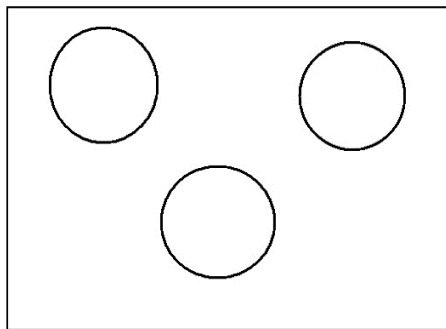
De forma similar a la obtención de las líneas, un círculo también puede representarse en el espacio bidimensional, sólo que en este caso necesita tres diferentes parámetros para su definición, los cuales son: el centro especificado por las dos coordenadas  $(u, v)$  y el radio  $\rho$ . La Figura 2.23 muestra los parámetros y su relación con la forma primitiva geométrica. En dicha figura, un punto  $\mathbf{p} = (x, y)$  forma parte del círculo si se cumple la siguiente condición:

$$\rho^2 = (x - u)^2 + (y - v)^2 \quad (2.29)$$

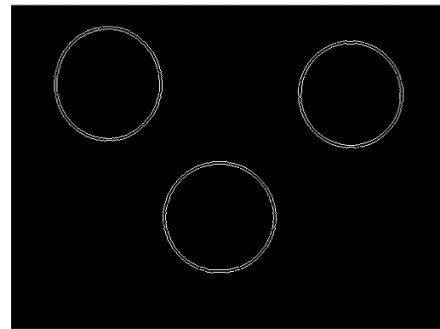


**Figura 2.23:** Parámetros que definen a un círculo.

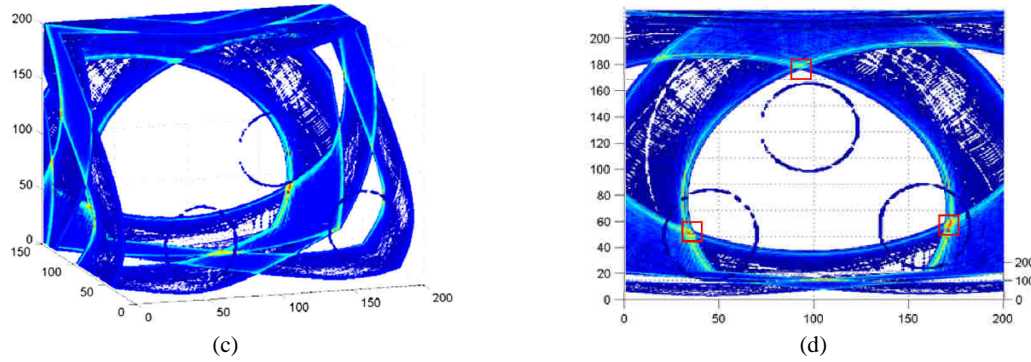
Considerando lo anterior, para implementar la transformada de Hough para detección de círculo es necesario disponer de un registro de acumuladores en tres dimensiones. Como primer paso se inicializan las celdas de acumulación del registro a cero. Después se realiza un recorrido de la imagen binaria, de tal forma que cada vez que se encuentre un píxel que representa a las líneas en la imagen binarizada, bien blanco o negro, se almacenan sus coordenadas  $(u, v)$  y se calcula el valor de  $\rho$ . Para cada conjunto de valores  $(u, v, \rho)$  obtenidos mediante dicho barrido, se incrementa el valor de la celda de acumulación del registro correspondiente a las coordenadas dadas. De esta forma, una vez que se han recorrido todos los píxeles de la imagen, las celdas del registro tridimensional que aparezcan con valores máximos son las que definen los círculos en la imagen. En la Figura 2.24 se muestra gráficamente el proceso.



(a)



(b)



**Figura 2.24:** Resultados parciales tras aplicar la transformada de Hough para detección de círculos: (a) imagen original; (b) bordes de (a); (c) representación tridimensional del arreglo de registros acumuladores; (d) detección de los máximos registrados.

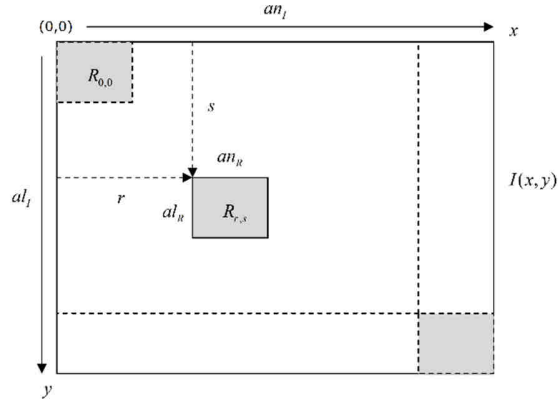
### 2.5.5 Comparación y reconocimiento de imágenes

En esta sección se analiza el problema de la comparación de imágenes o bien la localización de una parte conocida de la imagen, la cual normalmente se identifica como patrón o plantilla. Este tipo de problema es típico en aplicaciones tales como: en la búsqueda de puntos de referencia en visión estereoscópica, en la localización de un determinado objeto en una escena, o bien en el seguimiento de objetivos en una secuencia de imágenes. La idea fundamental de la comparación de imágenes (*template matching*, TM) es sencilla: el patrón o plantilla (*template*) a encontrar en la imagen se mueve sobre todos los píxeles de la imagen como si fuera un filtro lineal y se mide la semejanza entre el patrón y los correspondientes píxeles de la imagen que abarca en la exploración. Evidentemente la determinación de semejanza entre el patrón y cada uno de los puntos de la imagen no es sencilla, ya que debe considerarse que esta medida de semejanza sea fiable y robusta en casos tales como que el patrón se encuentra en la imagen escalado, rotado o distorsionado.

#### 2.5.5.1 Comparación en imágenes a escala de grises

Primero se abordará el problema de encontrar una imagen de referencia  $R(i, j)$  (patrón o plantilla) en una imagen a escala de grises  $I(x, y)$ . La tarea es encontrar el píxel  $(u, v)$  de la imagen  $I(x, y)$  en el cual exista una coincidencia óptima entre los contenidos del segmento de  $I(x, y)$  y de la imagen de referencia  $R(i, j)$ . Si se define  $R_{r,s}(i, j) = R(i - r, j - s)$  como un desplazamiento  $(r, s)$  en el sentido horizontal y vertical, entonces podría describirse el problema de la comparación de imágenes en escala de grises

según se expresa a continuación: definiendo  $I(x, y)$  como la imagen que contiene probablemente un determinado patrón  $R(i, j)$ , se debe encontrar el desplazamiento  $(r, s)$  en donde la medida de semejanza entre el patrón  $R(i, j)$  y el segmento de imagen que cubre  $R(i, j)$  es máxima (véase la Figura 2.25).



**Figura 2.25:** Geometría del problema de comparación de imágenes.

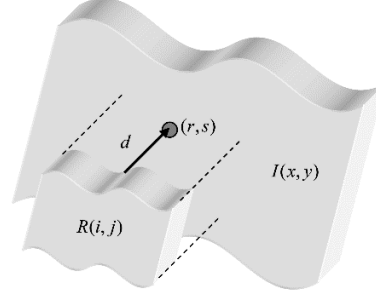
En la Figura 2.25 la imagen de referencia  $R(i, j)$  es desplazada a través de la imagen  $I(x, y)$ , donde el origen de la imagen  $(0,0)$  sirve como punto de referencia. El tamaño de la imagen  $an_I \times al_I$  y de la plantilla  $an_R \times al_R$ , determinan el campo de búsqueda para poder realizar la comparación.

Para poder desarrollar un algoritmo que resuelva el problema anterior es necesario tratar con tres situaciones importantes. Primero, se necesita la definición apropiada de una medida de semejanza entre las dos partes de la imagen. Segundo, se requiere una estrategia de búsqueda donde el desplazamiento óptimo  $(r, s)$  se encuentre de la manera más rápida posible. Finalmente y tercero, se debe decidir cuál es el valor mínimo de semejanza que asegura una coincidencia fiable.

### 2.5.5.2 Distancia entre patrones

Para calcular el punto donde existe la máxima coincidencia entre  $I(x, y)$  y  $R(i, j)$ , es necesario determinar la distancia entre la referencia desplazada  $R_{r,s}$  y la sección correspondiente de la imagen para cada posición  $(r, s)$  (véase la Figura 2.26). Para medir la

distancia entre dos funciones bidimensionales existen diferentes definiciones utilizables, de las cuales se describen a continuación las más importantes.



**Figura 2.26:** Medición de la distancia  $d$  entre funciones de dos dimensiones. La imagen de referencia está posicionada sobre el píxel  $I(r, s)$ .

Suma de las diferencias:

$$d_A(r, s) = \sum_{(i,j) \in R} |I(r+i, s+j) - R(i, j)| \quad (2.30)$$

El máximo de las diferencias:

$$d_M(r, s) = \max_{(i,j) \in R} |I(r+i, s+j) - R(i, j)| \quad (2.31)$$

Suma de las distancias cuadráticas:

$$d_E(r, s) = \sqrt{\sum_{(i,j) \in R} (I(r+i, s+j) - R(i, j))^2} \quad (2.32)$$

### 2.5.5.3 Distancia y correlación

La distancia euclidiana definida en la ecuación (2.32) es especialmente importante debido a sus propiedades estadísticas. Para encontrar la máxima coincidencia entre la imagen de referencia  $R(i, j)$  y la imagen  $I(x, y)$  es suficiente minimizar el cuadrado de  $d_E$ , el cual en todo caso se define como positivo. Lo anterior puede ser descrito como:

$$\begin{aligned} d_E^2(r, s) &= \sum_{(i,j) \in R} (I(r+i, s+j) - R(i, j))^2 \\ &= \underbrace{\sum_{(i,j) \in R} (I(r+i, s+j))^2}_{A(r,s)} + \underbrace{\sum_{(i,j) \in R} (R(i, j))^2}_B - 2 \underbrace{\sum_{(i,j) \in R} I(r+i, s+j) \cdot R(i, j)}_{C(r,s)} \end{aligned} \quad (2.33)$$

La expresión  $B$  en la ecuación (2.33) representa la suma al cuadrado de todos los

valores de la imagen de referencia  $R(i, j)$ . Como estos valores no dependen del desplazamiento  $(r, s)$ , este factor es constante en lo que se refiere al procesamiento de dicha ecuación. Considerando lo anterior  $B$  podría ignorarse desde el punto de vista de la minimización. La expresión  $A(r, s)$  representa la suma de los valores de los píxeles de la imagen correspondientes a  $R(i, j)$  al cuadrado. El valor  $A(r, s)$  claramente es dependiente del desplazamiento  $(r, s)$ .  $C(r, s)$  corresponde a la denominada correlación cruzada entre  $I(x, y)$  y  $R(i, j)$ . Dicha correlación cruzada se define en el caso general como:

$$(I \otimes R)(r, s) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(r+i, s+j) \cdot R(i, j) \quad (2.34)$$

Debido a que las imágenes  $I(x, y)$  y  $R(i, j)$ , más allá de sus dimensiones son consideradas como si contuvieran valores cero, la ecuación (2.34) puede reformularse como:

$$\sum_{i=0}^{an_R-1} \sum_{j=0}^{al_R-1} I(r+i, s+j) \cdot R(i, j) = \sum_{(i,j) \in R} I(r+i, s+j) \cdot R(i, j) \quad (2.35)$$

La correlación es fundamentalmente la misma operación que la convolución excepto que en la convolución el núcleo de la operación se encuentra invertido sobre sí mismo.

Si el factor  $A(r, s)$  de la ecuación (2.33) se mantiene aproximadamente constante dentro de toda la imagen  $I(x, y)$ , significaría que la energía de la imagen se encuentra distribuida uniformemente. En estas condiciones el valor máximo de la correlación  $C(r, s)$  corresponde al punto de la imagen  $I(x, y)$  donde la imagen de referencia  $R(i, j)$  tiene máxima coincidencia. En este caso el valor mínimo  $d_E^2(r, s)$  puede calcularse solamente mediante el valor máximo de la correlación  $I \otimes R$ . Lo anterior es algo muy importante ya que mediante la transformada de Fourier resulta altamente eficiente calcular la correlación en el dominio de la frecuencia.

#### 2.5.5.4 Correlación cruzada normalizada

En la práctica, la consideración de que  $A(r, s)$  se mantiene aproximadamente constante en la imagen no se cumple, por lo que el resultado de la correlación es altamente

dependiente de los cambios de intensidad de la imagen  $I(x, y)$ . La correlación cruzada normalizada  $C_N(r, s)$  compensa esta dependencia, en la cual la energía total de la imagen debe ser considerada. La correlación cruzada normalizada se define como:

$$C_N(r, s) = \frac{C(r, s)}{\sqrt{A(r, s) \cdot B}} = \frac{C(r, s)}{\sqrt{A(r, s) \cdot \sum_{(i, j) \in R} R(i, j)^2}} \quad (2.36)$$

$$= \frac{\sum_{(i, j) \in R} I(r+i, s+j) \cdot R(i, j)}{\sqrt{\sum_{(i, j) \in R} (I(r+i, s+j))^2 \cdot \sum_{(i, j) \in R} (R(i, j))^2}}$$

Como los valores tanto de la imagen  $I(x, y)$  como de la referencia  $R(i, j)$  son positivos, entonces los valores resultantes se encuentran dentro del intervalo  $[0, 1]$ . Un valor de  $C_N(r, s) = 1$  muestra un índice máximo de semejanza entre  $R(i, j)$  y la sección de la imagen  $I(x, y)$  con la que se corresponde. El valor que arroja la correlación cruzada normalizada tiene la ventaja de que puede ser utilizado como índice de semejanza estándar.

La formulación de la ecuación (2.36), a diferencia de la ecuación (2.34), muestra la caracterización de una medida de distancia local entre la distancia absoluta entre la imagen de referencia y la sección de la imagen  $I(x, y)$  con la que se corresponde. Sin embargo, un aumento en la iluminación general de la imagen produce cambios dramáticos en los valores de  $C_N(r, s)$ .

#### 2.5.5.5 Coeficiente de correlación

Una posibilidad para evitar este problema consiste en considerar, durante el procesamiento, en lugar de los valores de intensidad de los píxeles las diferencias locales de estos valores en relación a sus promedios. Considerando lo anterior la ecuación (2.36) se reformula como:

$$C_L(r, s) = \frac{\sum_{(i, j) \in R} (I(r+i, s+j) - \bar{I}(r, s)) \cdot (R(i, j) - \bar{R})}{\sqrt{\sum_{(i, j) \in R} (I(r+i, s+j) - \bar{I}(r, s))^2 \cdot \underbrace{\sum_{(i, j) \in R} (R(i, j) - \bar{R})^2}_{\sigma_R^2}}} \quad (2.37)$$

donde  $\bar{I}(r, s)$  y  $\bar{R}$  se definen como:

$$\bar{I}(r, s) = \frac{1}{N} \sum_{(i, j) \in R} I(r + i, s + j), \quad \bar{R} = \frac{1}{N} \sum_{(i, j) \in R} R(i, j) \quad (2.38)$$

donde  $N$  corresponde al número de elementos de la imagen de referencia  $R(i, j)$ . La ecuación (2.37) se conoce en el campo de la estadística como el coeficiente de correlación. El coeficiente de correlación no puede ser considerado como un índice de correlación global que considera el conjunto general de datos, sino como un factor local que considera sólo secciones concretas determinadas explícitamente por el tamaño de la imagen de referencia. El intervalo de valores en los que fluctúa el coeficiente de correlación es de  $[-1, 1]$ , donde el valor de 1 implica el mayor índice de semejanza, mientras que -1 indicaría que son totalmente diferentes  $I(x, y)$  y  $R(i, j)$ .

El denominador de la ecuación (2.37) contiene la expresión:

$$\sigma_R^2 = \sum_{(i, j) \in R} (R(i, j) - \bar{R})^2 = \sum_{(i, j) \in R} (R(i, j))^2 - N \cdot \bar{R}^2 \quad (2.39)$$

Esta expresión corresponde a la varianza del patrón  $R(i, j)$ . Dicho valor es constante en el procesamiento de los valores de  $C_L(r, s)$  por lo que solamente debe ser calculado una sola vez. A través de la sustitución de este factor en la ecuación (2.37), la expresión se reformularía como:

$$C_L(r, s) = \frac{\sum_{(i, j) \in R} (I(r + i, s + j) \cdot R(i, j)) - N \cdot \bar{I}(r, s) \cdot \bar{R}}{\sqrt{\sum_{(i, j) \in R} (I(r + i, s + j))^2 - N \cdot (\bar{I}(r, s))^2 \cdot \sigma_R}} \quad (2.40)$$

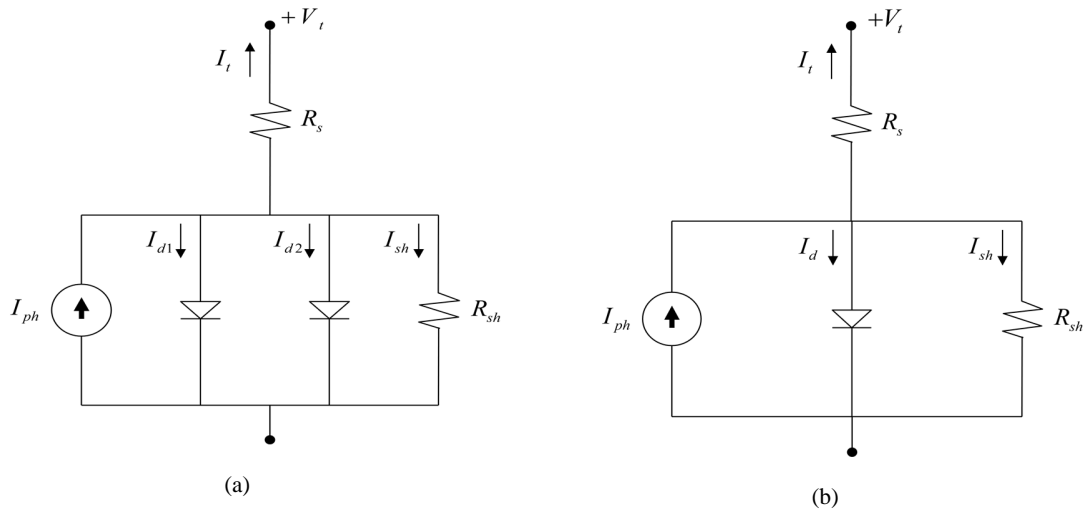
La expresión anterior corresponde a la manera más eficiente de calcular el coeficiente de correlación local. La eficiencia se debe a que  $\bar{R}$  y  $\sigma_R$  sólo deben ser calculados una única vez, mientras que el valor de  $\bar{I}(r, s)$  ya no participa en el procesamiento para cada píxel de la sección de la imagen, sino que ahora sólo es considerado una única vez.

El coeficiente de correlación calculado a partir de la ecuación (2.40) corresponde a un índice de medida local de la imagen a diferencia de la correlación lineal mostrada en la ecuación (2.34). Debido a que se trata de un índice local, no es posible establecer un método en el dominio de la frecuencia que calcule el mismo índice, ya que las técnicas espectrales formulan el cálculo de características globales.



## 2.6 Diseño y modelado de celdas solares

El modelado de celdas solares o fotovoltaicas consta de dos pasos: la definición del modelo matemático y una precisa estimación de sus parámetros. Una de las características más importantes para el modelo matemático es la relación corriente – voltaje ( $I$ - $V$ ). Partiendo de esto, el objetivo es encontrar la configuración que proporcione la mayor potencia, en las condiciones de operación deseadas. Se han propuesto diversos modelos en la literatura, sin embargo muchos de ellos debido a su complejidad resultan difíciles de implementar. En términos prácticos, se usan los modelos de diodo simple y doble [89].



**Figura 2.27:** Modelos de celdas solares: (a) modelo de doble diodo y (b) modelo de diodo simple.

Dentro del modelo de doble diodo, se considera que idealmente se tiene una fuente de corriente foto-generada, la cual pasa a través de un diodo rectificador. Sin embargo, en la práctica para el modelado de las celdas solares se deben considerar los aspectos detallados involucrados en el proceso de transformación de la energía. Por tal motivo, en el modelo del diodo doble se incorpora un diodo adicional al de la corriente foto-generada y un par de resistencias que modelan los contactos metálicos que posee la celda solar [164]. Si dentro del modelo de la celda solar sólo se maneja el diodo de la fuente de corriente foto generada, se considera diodo simple. La Figura 2.27 presenta los circuitos correspondientes al modelo de diodo doble y diodo simple. Por otra parte la ecuación que define el modelo de diodo simple es:

$$I_t = I_{ph} - I_{sd} \left[ \exp \left( \frac{q(V_t + R_s \cdot I_t)}{n \cdot k \cdot T} \right) - 1 \right] - \frac{V_t + R_s \cdot I_t}{R_{sh}} \quad (2.41)$$

y para el modelo de doble diodo:

$$I_t = I_{ph} - I_{sd1} \left[ \exp \left( \frac{q(V_t + R_s \cdot I_t)}{n_1 \cdot k \cdot T} \right) - 1 \right] - I_{sd2} \left[ \exp \left( \frac{q(V_t + R_s \cdot I_t)}{n_2 \cdot k \cdot T} \right) - 1 \right] - \frac{V_t + R_s \cdot I_t}{R_{sh}} \quad (2.42)$$

para ambos casos  $I_t$  es la corriente de la terminal,  $I_{ph}$  es la corriente foto-generada,  $I_{sd1}$  e  $I_{sd2}$  son las corrientes de difusión y saturación respectivamente,  $V_t$  es el voltaje en la terminal,  $R_s$  y  $R_{sh}$  son la resistencia en serie y la resistencia de desvío. Por otra parte, se manejan valores constantes que forman parte de la ecuación de Shockley para el diodo,  $q = 1.602 \times 10^{-19}$  culombios es la carga del electrón,  $k = 1.380 \times 10^{-23} (J/^{\circ}K)$  es la constante de Boltzmann,  $n_1$  y  $n_2$  son los coeficientes de difusión y recombinación, finalmente  $T$  es la temperatura de operación de la celda solar en  $^{\circ}K$ . En ambos modelos existen parámetros cuyo valor se desconoce ( $R_s$ ,  $R_{sh}$ ,  $I_{ph}$ ,  $I_{sd1}$ ,  $I_{sd2}$ ,  $n_1$  y  $n_2$ ), se han propuesto diversos métodos de optimización para su correcta elección [89, 91, 98, 102, 165-170], algunos de ellos se muestran en la Tabla 2.5.

## 2.7 Resumen

En este capítulo se realizó un estudio acerca de la teoría de los temas de algoritmos metaheurísticos, procesamiento de señales, procesamiento de imágenes y diseño de celdas solares. Dicho estudio sirve como base para proponer soluciones novedosas a los problemas planteados en el Capítulo 1 y así cumplir los objetivos propuestos. Además de estudiar los fundamentos de tales áreas, se presentaron los avances más recientes dentro de cada una. Para los fines de la investigación que se está desarrollando, no toda la información que se presentó es utilizada, sin embargo es necesario conocer los antecedentes para comprender los problemas y poder proponer las soluciones planteadas.

Nombre del método	Autores	Método de estimación
<i>Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers</i>	Easwarakhanthan et, al.	Métodos numéricos clásicos.
<i>Numerical method for the extraction of photovoltaic module double-diode model parameters through cluster analysis</i>	Sandrolini et, al.	Clustering y gradiente.
<i>A radial basis function neural network based approach for the electrical characteristics estimation of a photovoltaic module</i>	Bonanno et, al.	Redes neuronales.
<i>A Parameter extraction of solar photovoltaic modules using penalty-based differential evolution</i>	Ishaque et, al.	Algoritmos metaheurísticos.
<i>Parameter extraction of solar cells using particle swarm optimization</i>	Ye et, al.	Algoritmos metaheurísticos.
<i>A new estimation approach for determining the I–V characteristics of solar cells</i>	AlRashidi et, al.	Algoritmos metaheurísticos y búsqueda de patrones.
<i>Solar cell parameter extraction using genetic algorithms</i>	Jervase et, al.	Algoritmos metaheurísticos.
<i>Simulated Annealing algorithm for photovoltaic parameters identification</i>	El-Naggar et, al.	Algoritmos metaheurísticos
<i>Bacterial Foraging Algorithm based solar PV parameter estimation</i>	Rajasekar et, al.	Algoritmos metaheurísticos
<i>Parameter identification for solar cell models using harmony search-based algorithms</i>	Askarzadeh y Rezazadeh.	Algoritmos metaheurísticos

**Tabla 2.5:** Algunos métodos para el modelado y diseño de celdas solares.

Dentro de los algoritmos metaheurísticos y el aprendizaje basado en oposición se empleara la teoría explicada que permite realizar la selección de dos métodos de optimización metaheurísticos, basados en población llamados *Electromagnetism-Like Optimization* y *Harmony Search*. Estos algoritmos cumplen con la teoría explicada, además de esto para realizar la modificación de uno de ellos se hace uso de la teoría del aprendizaje basado en lo opuesto, en específico de las aportaciones realizadas por Tizhoosh et, al. Por otra parte en el tema de procesamiento de señales digitales, sólo se tendrá en cuenta la teoría planteada referente a los filtros IIR y las respectivas técnicas de diseño, de esta forma se puede elegir la que mejor se adapta al enfoque de optimización que se plantea. En el tema de procesamiento de imágenes se emplean las medidas de distancia, la cual servirá para poder definir diversos parámetros en la detección de objetos, los histogramas y en específico las técnicas de umbralizado de Otsu y Kapur son una gran referencia para el enfoque propuesto para segmentación basa en optimización. Mientras que los rasgos geométricos (área y perímetro) serán de utilidad para definir una nueva función objetivo que permita la detección de células sanguíneas. Para este mismo fin también es usada la transformada de Hough para detección de círculos, la cual sirve como referencia comparativa. Del mismo modo, para la detección y localización de objetos mediante el uso de plantillas o patrones se hace uso de las métricas explicadas, tanto la distancia, como correlación cruzada, constituyendo el punto de partida para dar solución a este problema. Para poder abordar el tema de diseño de celdas solares se toma en cuenta la teoría explicada, la cual introduce los modelos de diodo doble y simple, para las consideraciones dentro del enfoque de optimización se toman como referencia las propuestas realizadas Ishaque et, al. (DE), Ye et, al. (PSO), Jervase et, al. (GA), El-Naggar et, al. (SA), Rajasekar et, al. (BFA) y finalmente Askarzadeh y Rezazadeh que emplean HSA.



## Capítulo 3

# Algoritmo de optimización *Electromagnetism-Like*: modificaciones y aplicaciones

---

Este capítulo presenta al algoritmo de optimización *Electromagnetism – Like* (EMO). Primeramente, se hace una descripción de sus pasos básicos y de cada uno de los operadores que son usados en el proceso de optimización. Seguidamente se introduce la teoría que abarca la técnica llamada de aprendizaje basado en oposición (*Opposition-based learning*, OBL), para después mostrar la implementación de ambos métodos (EMO y OBL) como una mejora para el algoritmo EMO original. Para tal implementación son también mostrados los resultados experimentales.

Por otra parte, se muestran las implementaciones del algoritmo EMO para la identificación y modelado de filtros digitales, la detección de glóbulos blancos en imágenes médicas y finalmente, para localización y detección de objetos en imágenes digitales. Para estas implementaciones, primero se analiza la teoría sobre los problemas a resolver, seguido

de la aplicación y adaptación del algoritmo EMO y por último se muestran y valoran los resultados experimentales.

### 3.1 Algoritmo de optimización *Electromagnetism – Like* (EMO)

El algoritmo EMO es un método de optimización inspirado en procesos físicos. Este algoritmo permite la optimización global de funciones multimodales. En contraposición a los Algoritmos Genéticos (GA), EMO no emplea operadores de mutación o cruzamiento (*crossover*) para explorar el espacio de búsqueda, ya que se basa en el fenómeno físico del electromagnetismo, lo cual le permite tener operadores diferentes que trabajan directamente en el espacio de los números reales.

El algoritmo EMO permite resolver problemas de optimización definidos de la siguiente forma:

$$\text{Minimizar } f(\mathbf{x}), \quad \mathbf{x} \in [l, u] \quad (3.1)$$

donde  $l$  y  $u$  son los límites inferior y superior respectivamente del espacio de búsqueda  $\mathfrak{R}^n$  y se definen como:  $[l, u] = \{\mathbf{x} \in \mathfrak{R}^d \mid l_k \leq x_k \leq u_k, k=1, 2, \dots, d\}$ ,  $d$  es la dimensión del vector  $\mathbf{x}$  y finalmente  $f(\cdot)$  es la función a optimizar. Entonces es posible decir que lo descrito anteriormente son las características del problema a optimizar. Teniendo en cuenta estas consideraciones, el algoritmo EMO emplea dos procesos básicos para la optimización: el primero realiza una exploración del espacio de búsqueda de forma aleatoria, mientras que el segundo proceso consiste en la explotación local de los puntos elegidos, para este propósito EMO usa operadores basados en los principios de la teoría del electromagnetismo. Utilizando ambos procesos de forma balanceada se garantiza que el algoritmo converge en los valores mínimos (u óptimos) de la función que son altamente atrayentes, y se aleja de los valores máximos (no óptimos). Partiendo de estos procesos clave y en base a lo explicado en el Capítulo 2, el algoritmo EMO consta de cuatro fases para lograr la optimización global [13]. Cada una de dichas etapas se describe a continuación:

- **Inicialización:** un conjunto de  $m$  partículas son tomadas aleatoriamente considerando un espacio  $\mathfrak{R}^d$  definido por el límite superior ( $u$ ) y el límite inferior ( $l$ ).

- **Búsqueda local:** se realiza la búsqueda de un valor mínimo en la vecindad de un punto  $\mathbf{x}^p$ , donde  $p \in (1, 2 \dots m)$  y  $m$  corresponde al número total de individuos en la población.
- **Cálculo del vector de fuerza total:** en base al valor de la función objetivo se calculan las cargas y fuerzas para cada elemento de población de partículas.
- **Movimiento:** cada partícula de población es desplazada de acuerdo a la fuerza total calculada en base al valor de la función objetivo.

Cada uno de los elemento de EMO son descritos y analizados en cada pseudocódigo que se incluye a continuación.

*Inicialización:* se produce de manera aleatoria a partir de un grupo  $\mathbf{X}$  de  $m$  elementos, considerados como futuras soluciones, pero en este momento son sólo soluciones iniciales. Cada elemento de la población es considerada como una solución, o bajo el enfoque EMO una partícula cargada, y se asume que todas las partículas de la población están distribuidas uniformemente dentro del espacio de búsqueda. La mejor partícula o solución se encuentra al evaluar la función objetivo y ésta depende de cada problema de optimización. El proceso de inicialización finaliza cuando todas las  $m$  partículas de la población son evaluadas en la función objetivo, cabe señalar que de acuerdo al Capítulo 2 y los autores de EMO podemos definir la inicialización de acuerdo al Algoritmo 3.1.

---

**Algoritmo 3.1: Inicialización**


---

```

1: for  $i=1$  to  $m$  do
2:   for  $k=1$  to  $d$  do
3:      $\lambda \leftarrow rand(0,1)$ 
4:      $x_k^i \leftarrow l_k + \lambda(u_k - l_k)$ 
5:   end for
6: Calcular  $f(\mathbf{x}^i)$ 
7: end for

```

---

*Búsqueda local:* en esta etapa se intenta mejorar la solución ya encontrada tras evaluar toda la población en la función objetivo; sin embargo, para algunos problemas puede modificarse. Considerando esto, es posible formular una clasificación de los algoritmos basados en EMO: a) EMO sin búsqueda local; b) EMO con búsqueda local aplicada únicamente a la mejor partícula actual de población y c) EMO con búsqueda local aplicada todas las partículas, el cual es uno de los principales objetivos de estudio en este trabajo de investigación. La búsqueda local, trabaja considerando un determinado número de evaluaciones de la función objetivo  $LSITER$  y un parámetro de vecindad de búsqueda



( $\delta$ ). Este procedimiento funciona de la siguiente manera: el punto  $\mathbf{x}^p \in \mathbf{X}$  es asignado a una variable temporal, es necesario hacer uso de una variable temporal llamada  $\mathbf{y}$  para almacenar el valor principal (óptimo) de la población. Después de esto, para una coordenada dada  $d$  de una partícula, se selecciona un número aleatorio uniformemente distribuido ( $\lambda_1$ ), combinándose con  $\delta$ , así se obtiene la longitud de paso de búsqueda o vecindad. El elemento almacenado en  $\mathbf{y}$  se desplaza en la dirección que la longitud de paso indique, el signo que esta dirección tendrá también se determina de forma aleatoria ( $\lambda_2$ ). Si el valor obtenido tras evaluar  $\mathbf{y}$  en la función objetivo es mejor tras realizar *LSITER* iteraciones el punto  $\mathbf{x}^p$  es reemplazado por  $\mathbf{y}$ , terminando así la búsqueda en la vecindad de la dimensión analizada, de otra forma  $\mathbf{x}^p$  conservará su valor. Finalmente, el mejor punto actual es actualizado en la población, el pseudocódigo de este método se describe en el Algoritmo 3.2.

---

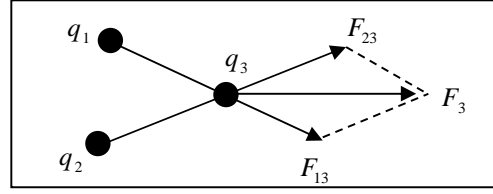
**Algoritmo 3.2: Búsqueda Local**


---

1: $\text{contador} \leftarrow 1$ 2: $\text{longitud} \leftarrow \delta(\max(u_k - l_k))$ 3: <b>for</b> $i = 1$ <b>to</b> $m$ <b>do</b> 4: <b>for</b> $k = 1$ <b>to</b> $d$ <b>do</b> 5: $\lambda_1 \leftarrow \text{rand}(0,1)$ 6: <b>while</b> $\text{contador} < \text{LSITER}$ <b>do</b> 7: $\mathbf{y} \leftarrow \mathbf{x}^i$ 8: $\lambda_2 \leftarrow \text{rand}(0,1)$ 9: <b>if</b> $\lambda_1 > 0.5$ <b>then</b> 10: $y_d = y_d - \lambda_2 \cdot (\text{Longitud})$ 11: <b>else</b>	12: $y_d = y_d + \lambda_2 \cdot (\text{Longitud})$ 13: <b>end if</b> 14: <b>if</b> $f(\mathbf{y}) < f(\mathbf{x}^i)$ <b>then</b> 15: $\mathbf{x}^p \leftarrow \mathbf{y}$ 16: $\text{contador} \leftarrow \text{LSITER} - 1$ 17: <b>end if</b> 18: $\text{contador} \leftarrow \text{contador} + 1$ 19: <b>end while</b> 20: <b>end for</b> 21: <b>end for</b> 22: $\mathbf{x}^{\text{mejor}} \leftarrow \arg \min(f(\mathbf{x}^i), \mathbf{x}^i \in \mathbf{X})$
--	--

---

De forma general, la búsqueda local aplicada a todas las partículas puede reducir el riesgo de caer en un mínimo local, pero la desventaja es que tiende a aumentar el tiempo de cómputo. Mantener la búsqueda local centrada en la mejor partícula actual resulta más conveniente, ya que así es posible conservar la eficiencia y precisión computacional. Sin embargo esto reduce la explotación del espacio de búsqueda. En la búsqueda local, la longitud de paso representa un importante factor que depende de los límites de cada dimensión y determina el desempeño del método de dicha búsqueda.

**Figura 3.1:** Principio de superposición.

*Calculo del vector de fuerza total:* El cálculo del vector de fuerza total se basa en el principio de superposición (Figura 3.1) de la teoría del electromagnetismo ecuación (3.1), el cual establece que: “la fuerza ejercida en una partícula por medio de otra partícula es inversamente proporcional a la distancia entre los puntos y directamente proporcional al producto de sus cargas” [171]. La ecuación (3.1) es un ejemplo del principio de superposición para obtener la fuerza total ejercida en una carga de acuerdo al sistema de cargas de la Figura 3.1.

$$\vec{F}_{i,3} = \left( \frac{q_3 \cdot q_i}{4\pi\epsilon_0\epsilon_r r^2} \right) \cdot \vec{e}_r, \quad i = 1, 2 \quad (3.1)$$

De esta forma dentro del enfoque EMO, cada partícula se desplaza de acuerdo a la ley de Coulomb (Figura 3.2), la cual emplea la fuerza producida entre las partículas, que depende del valor de carga poseída por cada una de ellas. Dicha carga es determinada por su valor de desempeño en la función objetivo y se calcula de la siguiente manera:

$$q^p = \exp \left( -n \frac{f(\mathbf{x}^p) - f(\mathbf{x}^{mejor})}{\sum_{h=1}^m (f(\mathbf{x}^h) - f(\mathbf{x}^{mejor}))} \right), \forall p \quad (3.2)$$

donde  $n$  denota la dimensión de  $\mathbf{x}^p$  y  $m$  representa el tamaño de la población de partículas, el vector  $\mathbf{x}^h$  corresponde a la  $h$ -ésima partícula de la población.  $\mathbf{x}^p$  es el elemento actual de la población para el cual se está calculando la carga  $q$ ,  $\exp$  corresponde a la función exponencial y  $f(\cdot)$  corresponde a la función objetivo. Un problema de optimización con un número elevado de dimensiones generalmente requiere de una mayor población. En la ecuación (3.2)  $\mathbf{x}^{mejor}$  es la mejor partícula de acuerdo al valor de la función objetivo y es la que tiene una mayor carga.

La fuerza de atracción que ejerce la mejor partícula sobre otra partícula dada es inversamente proporcional a la distancia existente entre ellas. Por lo tanto, esta partícula atrae otras partículas que tengan peores resultados de afinidad y repele a las que tienen

mejores valores. La fuerza resultante que existe entre las partículas determina la modificación del valor de las partículas en el proceso de optimización. La fuerza de cada partícula se calcula por la ley de Coulomb y el principio de superposición, por medio de la ecuación (3.3).

$$\mathbf{F}^p = \sum_{h \neq p}^m \left\{ \begin{array}{ll} \left( \mathbf{x}^h - \mathbf{x}^p \right) \frac{q^p q^h}{\|\mathbf{x}^h - \mathbf{x}^p\|^2} & \text{si } f(\mathbf{x}^h) < f(\mathbf{x}^p) \\ \left( \mathbf{x}^p - \mathbf{x}^h \right) \frac{q^p q^h}{\|\mathbf{x}^h - \mathbf{x}^p\|^2} & \text{si } f(\mathbf{x}^h) \geq f(\mathbf{x}^p) \end{array} \right\}, \forall p \quad (3.3)$$

donde  $f(\mathbf{x}^h) < f(\mathbf{x}^p)$  representa la atracción y  $f(\mathbf{x}^h) \geq f(\mathbf{x}^p)$  representa la repulsión de cargas, un ejemplo de esto se aprecia en la Figura 3.2. La fuerza resultante de cada partícula es proporcional al producto de las cargas e inversamente proporcional a la distancia entre las partículas. Para que el proceso sea numéricamente consistente la ecuación (3.3) debe normalizarse como se expresa a continuación.

$$\overline{\mathbf{F}}^p = \frac{\mathbf{F}^p}{\|\mathbf{F}^p\|}, \quad \forall p \quad (3.4)$$

El Algoritmo 3.3 presenta los pasos necesarios para realizar el cálculo del vector de fuerza total.

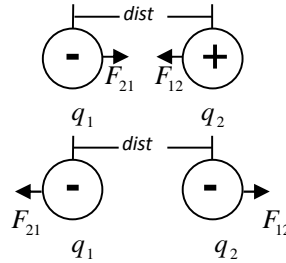


Figura 3.2: Ley de Coulomb.

---

**Algoritmo 3.3: Cálculo del vector de fuerza total**

---

```

1: for  $i=1$  to  $m$  do
2:    $q^i \leftarrow \exp \left( -n \frac{f(\mathbf{x}^i) - f(\mathbf{x}^{mejor})}{\sum_{h=1}^m (f(\mathbf{x}^h) - f(\mathbf{x}^{mejor}))} \right)$ 
3:    $\mathbf{F}^i \leftarrow 0$ 
4: end for
5: for  $i=1$  to  $m$  do
6:   for  $k=1$  to  $d$  do

```

---

```

7:          if  $f(\mathbf{x}^j) < f(\mathbf{x}^i)$  then
8:               $\mathbf{F}^i \leftarrow \mathbf{F}^i + (\mathbf{x}^j - \mathbf{x}^i) \frac{q^i q^j}{\|\mathbf{x}^j - \mathbf{x}^i\|^2}, \text{ (Atracción)}$ 
9:          else
10:              $\mathbf{F}^i \leftarrow \mathbf{F}^i - (\mathbf{x}^j - \mathbf{x}^i) \frac{q^i q^j}{\|\mathbf{x}^j - \mathbf{x}^i\|^2}, \text{ (Repulsión)}$ 
11:          end if
12:      end for
13: end for

```

---

*Movimiento:* De acuerdo a la fuerza resultante cada partícula se desplaza como indica la siguiente ecuación:

$$\mathbf{x}^p = \begin{cases} \mathbf{x}^p + \lambda \cdot \mathbf{F}^p \cdot (u_d - \mathbf{x}_d^p) & \text{si } \mathbf{F}^p > 0 \\ \mathbf{x}^p + \lambda \cdot \mathbf{F}^p \cdot (\mathbf{x}_d^p - l_d) & \text{si } \mathbf{F}^p \leq 0 \end{cases}, \forall p \neq \text{mejor} \quad (3.5)$$

donde  $\lambda$  es un valor aleatorio uniformemente distribuido entre  $[0,1]$ , por otro lado  $u_d$  y  $l_d$  son los límites superior e inferior de la dimensión  $d$ , respectivamente. Si la fuerza es repulsiva, la partícula  $\mathbf{x}^p$  se desplaza hacia el límite superior, en caso contrario el movimiento tiende hacia el límite inferior, en ambos casos empleando el valor aleatorio. La mejor de las partículas no se mueve, debido a que es la que tiene una atracción absoluta y repele o atrae a los demás elementos de la población.

El proceso termina cuando se alcanza un número máximo de iteraciones o cuando un valor  $f(\mathbf{x}^{\text{mejor}})$  es óptimo en algún sentido. Estas tres fases de EMO (búsqueda local, cálculo del vector de fuerza total y movimiento), representan el proceso de explotación para encontrar el valor óptimo. El Algoritmo 3.4 muestra cada uno de los pasos del proceso de movimiento de partículas.

---

**Algoritmo 3.4: Movimiento**

---

```

1: for  $i=1$  to  $m$  do
2:     if  $i \neq \text{mejor}$  then
3:          $\lambda \leftarrow \text{rand}(0,1)$ 
4:          $\mathbf{F}^i = \frac{\mathbf{F}^i}{\|\mathbf{F}^i\|}$ 
5:         for  $k=1$  to  $d$  do
6:             if  $F_k^i > 0$  then
7:                  $x_k^i \leftarrow x_k^i + \lambda \cdot F_k^i \cdot (u_k - x_k^i)$ 
8:             else
9:                  $x_k^i \leftarrow x_k^i + \lambda \cdot F_k^i \cdot (x_k^i - l_k)$ 
10:            end if
11:        end for
12:    end if

```

---

13: end for

### 3.2 Algoritmo *Opposition-based Learning* (OBL)

El aprendizaje basado en lo opuesto (*Opposition-based Learning*, OBL), propuesto por Tizhoos en [45], es un nuevo concepto en inteligencia computacional que es empleado de tal forma que existe una mejora en muchos algoritmos [172, 173]. Este enfoque de forma simultánea evalúa una solución  $\mathbf{x}^p$  y su valor opuesto  $\overline{\mathbf{x}}^p$  para un problema dado. Además provee una oportunidad distinta de encontrar una nueva solución candidata la cual puede estar cercana al óptimo global [46].

#### 3.2.1 Concepto de número opuesto

Dado un punto  $\mathbf{x}^p \in [l_d, u_d]$  como un número real, donde  $u_d$  y  $l_d$  representan el límite superior e inferior respectivamente. El número opuesto  $\overline{\mathbf{x}}^p$  se define como:

$$\overline{\mathbf{x}}^p = u_d + l_d - \mathbf{x}^p \quad (3.6)$$

#### 3.2.2 Concepto de un punto opuesto en el espacio

De forma similar a la definición de un número opuesto, es posible generalizar tal definición para más dimensiones. Dado un punto  $\mathbf{x}^p = (x_1, x_2, \dots, x_n)$  en un espacio  $n$ -dimensional donde cada componente  $x_1, x_2, \dots, x_d \in \Re$  y  $x_j \in [l_j, u_j]$ , donde  $j = 1, 2, \dots, d$ . El punto opuesto  $\overline{\mathbf{x}}^p = (\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)$  se calcula usando la siguiente ecuación:

$$\overline{\mathbf{x}}_j = u_j + l_j - \mathbf{x}_j \quad (3.7)$$

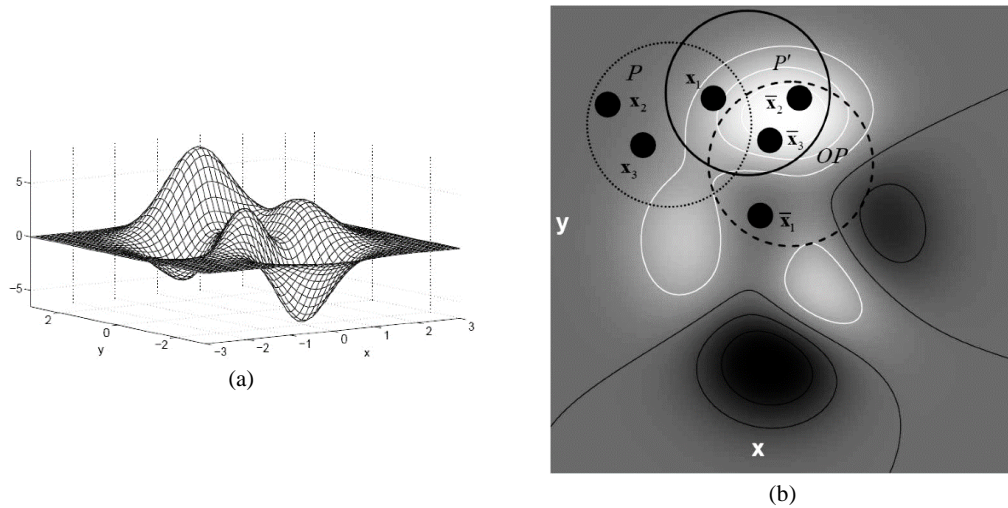
#### 3.2.3 Optimización basada en lo opuesto

Los métodos metaheurísticos usualmente comienzan su proceso considerando una población inicial de posibles soluciones y probando si éstas son capaces de mejorar hacia soluciones óptimas. La búsqueda finaliza cuando se alcanza un criterio de paro predefinido. Con la falta de información, a priori, sobre la solución óptima, se consideran supuestos aleatorios. Las diferencias en el tiempo de cómputo existentes entre diversos algoritmos se toman como una característica y tiene relación con la distancia que los valores iniciales poseen con respecto a la solución óptima. Las soluciones iniciales pueden tener la

oportunidad de comenzar la búsqueda en posiciones más cercanas al óptimo, si de forma simultánea se analiza el valor opuesto de cada una. Haciendo esto, la solución más cercana (opuesta o no) puede ser considerada como solución inicial. Esta suposición parte de la teoría de la probabilidad, donde el 50% de las veces una solución está más alejada que su valor opuesto [47]. La cercanía de un punto con respecto al óptimo global se define por medio de los valores de la función objetivo. Por lo tanto, empezando con el elemento más cercano de los dos, es posible acelerar la convergencia. El mismo enfoque no sólo puede ser aplicado a las soluciones iniciales, sino que también, puede usarse después de cada modificación de la población actual.

En base a la definición de un punto opuesto, el aprendizaje basado en lo opuesto se puede definir de la siguiente manera: dado un punto  $\mathbf{x}^p$  en un espacio  $n$ -dimensional. Se asume que  $f(\mathbf{x}^p)$  es la función objetivo, la cual evalúa la calidad de las soluciones. De acuerdo con la definición de un punto opuesto,  $\overline{\mathbf{x}^p}$  es el opuesto de  $\mathbf{x}^p$ . Si  $f(\overline{\mathbf{x}^p})$  es mejor que  $f(\mathbf{x}^p)$ , entonces  $\mathbf{x}^p$  se actualiza con el valor de  $\overline{\mathbf{x}^p}$  en otro caso el valor del punto actual se mantiene. Por lo tanto, el mejor punto ( $\mathbf{x}^p$  o  $\overline{\mathbf{x}^p}$ ) es modificado usando los operadores conocidos de alguno de los algoritmos de optimación basados en población.

La Figura 3.3 muestra el procedimiento de OBL. Las Figuras 3.3 (a) y 3.3 (b) representan la función a ser optimizada y su correspondiente gráfica de contornos, respectivamente.

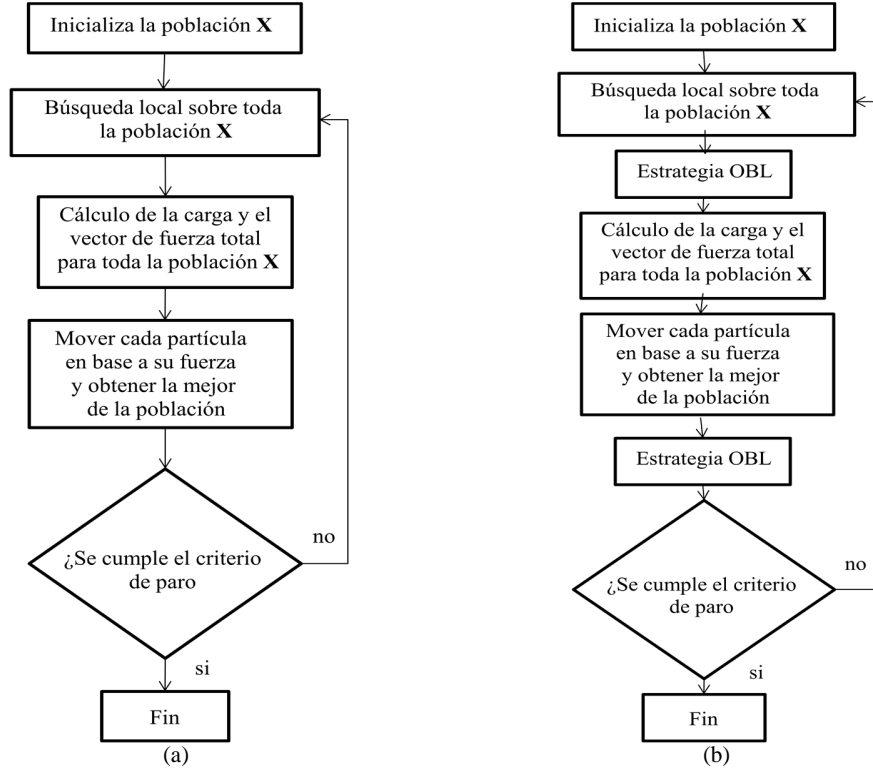


**Figura 3.3:** Procedimiento de la optimización basada en lo opuesto: (a) función a optimizar, (b) contorno de la función a optimizar.

Más específicamente, en la Figura 3.3 se presenta el siguiente ejemplo: la población en la iteración actual se define como  $\mathbf{P}$  y cuenta con tres partículas  $\mathbf{P} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$ , al aplicar los principios de OBL se genera una nueva población  $\mathbf{OP} = [\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \overline{\mathbf{x}}_3]$ . Los tres mejores elementos que existen en  $\mathbf{P}$  y  $\mathbf{OP}$  se seleccionan como una nueva población  $\mathbf{P}'$ . En la Figura 3.3 (b) se aprecia que los elementos de la nueva población son  $\mathbf{P}' = [\mathbf{x}_1, \overline{\mathbf{x}}_2, \overline{\mathbf{x}}_3]$ , en este caso el procedimiento muestra que  $\mathbf{x}_1$  de la población actual es el elemento más alejado del valor óptimo (máximo), por lo tanto no provee ninguna oportunidad de mejorar la población y será eliminado en la siguiente iteración.

### 3.3 *Opposition-Based Electromagnetism-Like Optimization (OBEMO)*

Como ya se mencionó en el Capítulo 2, la mayoría de los algoritmos metaheurísticos poseen dos pasos fundamentales y el algoritmo EMO no es la excepción, requiriéndose la inicialización de la población y la producción de nuevas generaciones por medio de operadores evolutivos. En este enfoque, el esquema de OBL se incorpora para mejorar ambos pasos. Sin embargo, el algoritmo EMO original se considera como el algoritmo principal y el procedimiento OBL sólo se incorpora para acelerar la convergencia. A continuación se presenta la descripción del método OBEMO seguido de los resultados experimentales obtenidos tras su aplicación.



**Figura 3.4:** Diagrama de flujo para: (a) algoritmo EMO y (b) OBEMO.

### 3.3.1 Implementación del algoritmo OBEMO

En la inicialización de las técnicas metaheurísticas basadas en población, comúnmente se emplean números aleatorios para crear una población inicial por la ausencia de conocimiento a priori del problema. Sin embargo, como se mencionó anteriormente es posible comenzar la optimización con mejores soluciones candidatas usando el aprendizaje basado en lo opuesto. Los siguientes pasos explican en detalle el proceso de inicialización empleado para el método propuesto.

- 1) Inicializar una población  $\mathbf{X}$  de  $N_p$  partículas usando el Algoritmo 3.1.
- 2) Calcular la población opuesta por medio de la siguiente ecuación:

$$\overline{x}_j^i = u_j + l_j - x_j^i \quad i = 1, 2, \dots, N_p; \quad j = 1, 2, \dots, d \quad (3.8)$$

donde  $x_j^i$  y  $\overline{x}_j^i$  denotan el  $j$ -ésimo parámetro de la  $i$ -ésima partícula de la población y su correspondiente partícula opuesta.

- 3) Selección de los  $N_p$  mejores elementos del conjunto de poblaciones  $\{\mathbf{X} \cup \overline{\mathbf{X}}\}$ , los cuales conforman parte de la población inicial.



Partiendo de la población actual, la estrategia OBL puede usarse de nuevo para producir nuevas poblaciones. En este procedimiento, la población opuesta es calculada y los mejores individuos son seleccionados como en el paso 3 de la inicialización. En la Figura 3.4 se muestra el diagrama de flujo tanto del EMO original como del OBEMO con la finalidad de comparar ambos métodos e identificar en qué punto se incorpora la estrategia OBL. Por otra parte, como continuación se describe el proceso de OBEMO paso a paso:

- Paso 1:** Generar  $N_p$  partículas aleatorias con dimensión  $d$ , que conforman la población  $\mathbf{X}$  (Algoritmo 3.1).
- Paso 2:** Aplicar la estrategia OBL considerando el vector  $\mathbf{X}$ , se genera el vector opuesto  $\bar{\mathbf{X}}$  usando la ecuación (3.8).
- Paso 3:** Seleccionar los  $N_p$  mejores elementos existentes entre la población actual y la opuesta  $\{\mathbf{X} \cup \bar{\mathbf{X}}\}$  de acuerdo a  $f(\bullet)$ , estas partículas construyen la población inicial  $\mathbf{X}_0$ .
- Paso 4:** Aplicar la búsqueda local descrita en el Algoritmo 3.2 a cada partícula de la población  $\mathbf{X}_0$ .
- Paso 5:** Calcular la carga entre las partículas usando la ecuación (3.2) y el vector de fuerza total por medio de la ecuación (3.3). La partícula con el mejor valor de la función objetivo tendrá una mayor carga y una mayor fuerza de atracción.
- Paso 6:** Desplazar las partículas de acuerdo a la magnitud de la fuerza. Esta nueva posición es calculada por la ecuación (3.4).  $\mathbf{x}^{mejor}$  no se desplaza porque tiene una mayor fuerza y atrae a otras partículas hacia ella. El resultado de este proceso se almacena en  $\mathbf{X}_{mov}$ .
- Paso 7:** Aplicar la estrategia OBL sobre el vector  $\mathbf{X}_{mov}$  y usando la ecuación (3.8) calcular  $\bar{\mathbf{X}}_{mov}$ .
- Paso 8:** Seleccionar los  $N_p$  mejores elementos existentes entre  $\{\mathbf{X}_{mov} \cup \bar{\mathbf{X}}_{mov}\}$  de acuerdo al valor de  $f(\bullet)$ . Tales partículas son almacenadas en  $\mathbf{X}_0$ .
- Paso 9:** Incrementar el índice de iteraciones y verificar si se cumple el criterio de parada. Si éste se cumple ir al Paso 10, en caso contrario ir al Paso 4.
- Paso 10:** La mejor partícula  $\mathbf{x}^{mejor}$  se extrae después de la última iteración, la cual es considerada como la solución global del problema.

### 3.3.2 Resultados experimentales

Con el propósito de comprobar el rendimiento de los algoritmos, el método propuesto OBEMO se compara con el EMO estándar, así como con otras modificaciones realizadas al algoritmo original que se encuentran dentro de la literatura, tal y como se describe posteriormente.

Para el estudio experimental se ha elegido un amplio conjunto de problemas de referencia, que incluye 14 diferentes pruebas de optimización global. De acuerdo a su uso en la literatura, dichas funciones se dividen en dos grupos diferentes: las funciones originales ( $f_1 - f_9$ ) y las funciones multidimensionales ( $f_{10} - f_{14}$ ). Todas las funciones son consideradas como problemas de minimización.

Las funciones originales que se presentan en la Tabla 3.1 concuerdan con el conjunto de funciones numéricas presentadas en el artículo original donde se propuso el algoritmo EMO [13]. Considerando esto, dicho grupo de funciones es usado también en la mayoría de las nuevas modificaciones basadas en EMO, ya que su uso facilita el estudio experimental y las comparaciones con trabajos similares.

El mayor reto de los enfoques que emplea EMO, es evitar la complejidad computacional que surgen debido a un gran número de iteraciones requeridas en la búsqueda local. Como dicha complejidad depende de la dimensión del problema a optimizar, se utiliza un conjunto de funciones multidimensionales (Tabla 3.2) con el propósito de comprobar la convergencia y precisión de cada técnica. Las funciones multidimensionales incluyen un grupo de cinco funciones distintas, cuyas dimensiones en el espacio de búsqueda se configuraron a  $d=30$ .

Función	Dominio de búsqueda	Mínimo global
<b>Branin</b> $f_1(x_1, x_2) = (x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	$-5 \leq x_1 \leq 10$ $0 \leq x_2 \leq 15$	0.397887
<b>Camel</b> $f_2(x_1, x_2) = -\frac{-x_1^2 + 4.5x_2^2 + 2}{e^{2x_1^2}}$	$-2 \leq x_1, x_2 \leq 2$	-1.031
<b>Goldstein-Price</b> $f_3(x_1, x_2) = 1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 13x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \times (30 + 2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$	$-2 \leq x_1, x_2 \leq 2$	3.0
<b>Hartmann (3-dimensional)</b> $f_4(\mathbf{x}) = -\sum_{i=1}^4 \alpha_i \exp\left[-\sum_{j=1}^3 A_{ij}(x_j - P_{ij})^2\right]$ $\alpha = [1, 1.2, 3, 3.2], \mathbf{A} = \begin{bmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 35 \end{bmatrix},$ $\mathbf{P} = 10^{-4} \begin{bmatrix} 6890 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix}$	$0 \leq x_i \leq 1$ $i=1,2,3$	-3.8627
<b>Hartmann (6-dimensional)</b> $f_5(\mathbf{x}) = -\sum_{i=1}^4 \alpha_i \exp\left[-\sum_{j=1}^6 B_{ij}(x_j - Q_{ij})^2\right]$ $\alpha = [1, 1.2, 3, 3.2], \mathbf{B} = \begin{bmatrix} 10 & 3 & 17 & 3.05 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix},$ $\mathbf{Q} = 10^{-4} \begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix}$	$0 \leq x_i \leq 1$ $i=1,2,3,\dots,6$	-3.8623
<b>Shekel <math>S_m</math> (4-dimensional)</b> $S_m(\mathbf{x}) = -\sum_{j=1}^m \left[ \sum_{i=1}^4 (x_i - C_{ij})^2 + \beta_j \right]^{-1}$ $\beta = [1, 2, 2, 4, 4, 6, 3, 7, 5, 5]^T,$ $\mathbf{C} = \begin{bmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 5.0 & 1.0 & 2.0 & 3.6 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 3.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \end{bmatrix}$	$0 \leq x_i \leq 10$ $i=1,2,3,4$	
$f_6(\mathbf{x}) = S_5(\mathbf{x})$		-10.1532
$f_7(\mathbf{x}) = S_7(\mathbf{x})$		-10.4029
$f_8(\mathbf{x}) = S_{10}(\mathbf{x})$		-10.5364
<b>Shubert</b> $f_9(x_1, x_2) = \left( \sum_{i=1}^s i \cos((i+1)x_1 + i) \right) \left( \sum_{i=1}^s i \cos((i+1)x_2 + i) \right)$	$-10 \leq x_1, x_2 \leq 10$	-186.73

Tabla 3.1: Funciones de optimización de prueba, correspondientes al conjunto de original.

Función	Dominio de búsqueda	Mínimo global
$f_{10}(\mathbf{x}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$x \in [-5.12, 5.12]^{30}$	0
$f_{11}(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20$	$x \in [-32, 32]^{30}$	0
$f_{12}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x \in [-600, 600]^{30}$	0
$f_{13}(\mathbf{x}) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4} \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$x \in [-50, 50]^{30}$	0
$f_{14}(\mathbf{x}) = \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)]$ $+ (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] + \sum_{i=1}^n u(x_i, 5, 100, 4)$	$x \in [-50, 50]^{30}$	0

**Tabla 3.2:** Conjunto de funciones de prueba multidimensionales.

El objetivo del conjunto experimental de pruebas es comparar cuatro algoritmos basados en EMO incluyendo el método propuesto OBEMO. Todos los algoritmos encaran los 14 problemas propuestos, dichos algoritmos se enlistan a continuación:

- Algoritmo EMO estándar [13].
- Algoritmo EMO híbrido con búsqueda descendiente (*Hybridizing EMO*, HEMO) [14]
- Algoritmo EMO con patrón de búsqueda definido (*Fixed pattern search EMO*, FEMO) [105].
- Algoritmo propuesto OBEMO.

Para el algoritmo EMO original descrito en [13] y el OBEMO la configuración de los parámetros de configuración son los siguientes:  $\delta = 0.001$  y  $LISTER = 4$ . Para HEMO se consideran los parámetros propuestos por los autores del propio método [14]:  $LsIT_{\max} = 10$ ,  $\varepsilon_r = 0.001$  y  $\gamma = 0.00001$ . A diferencia de los algoritmos EMO estándar y OBEMO, el método HEMO reduce las iteraciones en la fase de búsqueda local, aplicando tal proceso sólo a la mejor partícula encontrada. El conjunto de parámetros para el enfoque FEMO se define considerando:  $N_{fe}^{\max} = 100$ ,  $\delta = 0.001$ ,  $\delta^{\min} = 1 \times 10^{-8}$  y  $\varepsilon_{\delta} = 0.1$ . Todos los algoritmos mencionados utilizan el mismo tamaño de población  $m = N_p = 50$ .

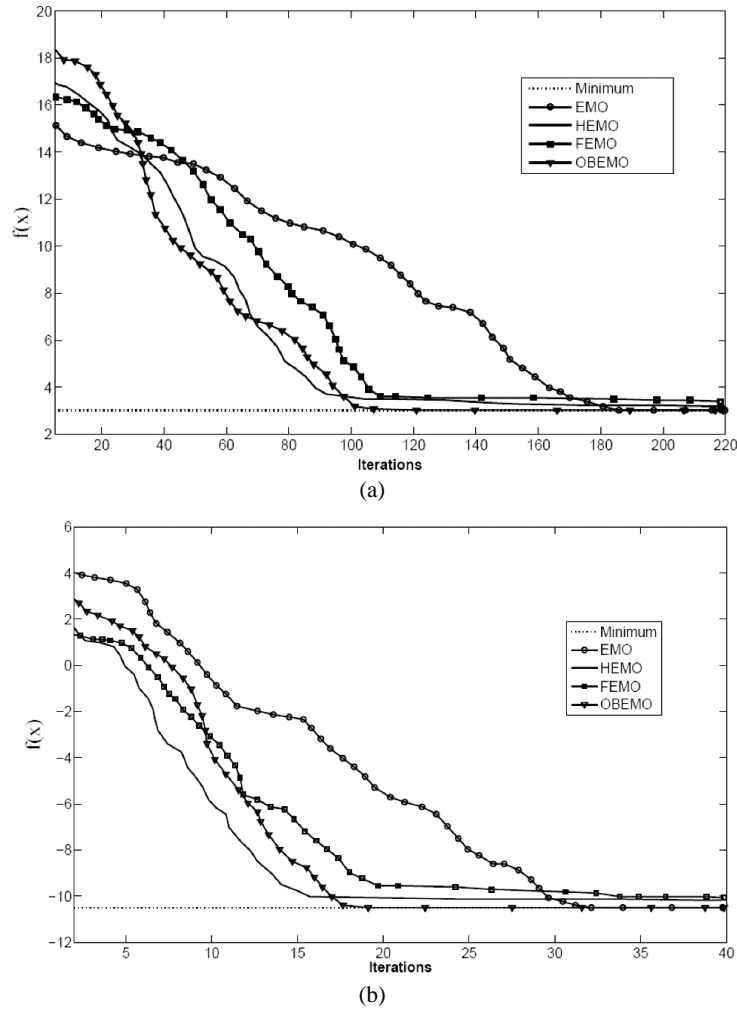
A continuación se presentan los resultados experimentales, que se dividen en base a los dos conjuntos de funciones de prueba.

*Funciones de prueba originales:* En este grupo de funciones se compara el rendimiento del algoritmo OBEMO con el de los métodos EMO, HEMO y FEMO, considerando las funciones de prueba con las que surgió el enfoque EMO. Dichas funciones son presentadas en la Tabla 3.1, las cuales tienen diferente número de dimensiones y un mínimo global conocido. El rendimiento se analiza considerando 35 diferentes ejecuciones para cada algoritmo. Como criterio de parada se considera el caso en el cual no exista un cambio significativo en la solución de cada iteración (e. g., un valor menor que  $10^{-4}$ ).

Los resultados se muestran en la Tabla 3.3, su evaluación se realiza asumiendo el promedio del mejor valor de la función objetivo  $f(\mathbf{x})$  y el promedio del mayor número de iteraciones ( $MAXITER$ ). En la Figura 3.5 se muestra el proceso de evolución para las funciones  $f_3$  y  $f_6$ , donde los valores de dichas funciones corresponden al mejor caso para cada enfoque obtenido. Para obtener el mejor valor de la función se ejecuta el algoritmo propuesto 35 veces para cada función a optimizar. Cada ejecución corresponde a un experimento que además de ayudar a identificar el mejor valor, sirve para evaluar estadísticamente el rendimiento del método propuesto, tal como se presenta más adelante.

Función		$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$
Dimensión		2	2	2	3	6	4	4	4	2
EMO	Promedio del mejor valor $f(x)$	0.3980	-1.015	3.0123	-3.7156	-3.6322	-10.07	-10.23	-10.47	-186.71
	Promedio de $MAXITER$	103	128	197	1.59E+03	1.08E+03	30	31	29	44
OBEMO	Promedio del mejor valor $f(x)$	0.3980	-1.027	3.0130	-3.7821	-3.8121	-10.11	-10.22	-10.50	-186.65
	Promedio de $MAXITER$	61	83	101	1.12E+03	826	18	19	17	21
HEMO	Promedio del mejor valor $f(x)$	0.5151	-0.872	3.413	-3.1187	-3.0632	-9.041	-9.22	-9.1068	-184.31
	Promedio de $MAXITER$	58	79	105	1.10E+03	805	17	18	15	22
FEMO	Promedio del mejor valor $f(x)$	0.4189	-0.913	3.337	-3.3995	-3.2276	-9.229	-9.88	-10.18	-183.88
	Promedio de $MAXITER$	63	88	98	1.11E+03	841	21	22	19	25

**Tabla 3.3:** Resultados comparativos para los algoritmos EMO, OBEMO, HEMO y FEMO considerando las funciones originales (Tabla 3.1).



**Figura 3.5:** Comparación del proceso de optimización para dos de las funciones de prueba originales: (a)  $f_3$  y (b)  $f_6$ .

Con el objetivo de analizar estadísticamente los resultados de la Tabla 3.3, se ha llevado a cabo una prueba no paramétrica conocida como la prueba de rangos de Wilcoxon (*Wilcoxon's rank test*) [174-176]. Dicha prueba permite evaluar las diferencias de resultados entre los dos métodos relacionados. El análisis se realiza considerando un nivel de significancia del 5% sobre los datos del “promedio del mejor valor de  $f(\mathbf{x})$ ” y “promedio del número máximo de iteraciones o *MAXITER*”. Las Tablas 3.4 y 3.5 reportan los valores  $p$  producidos por la prueba de Wilcoxon, para la comparación por pares de los promedios del mejor valor en la función objetivo y del número máximo de iteraciones, respectivamente. Se consideran tres grupos que están conformados por OBEMO vs. EMO, OBEMO vs. HEMO y OBEMO vs. FEMO. Como una hipótesis nula, se asume que no existe diferencia entre los valores de ambos algoritmos, la hipótesis alternativa considera la

existencia de una diferencia entre ambos enfoques. Los resultados obtenidos por la prueba de Wilcoxon indican que las soluciones analizadas no ocurren por coincidencia, tal como podría ser ruido normal contenido en el proceso.

Función	Valores $p$		
	OBEMO vs. EMO	OBEMO vs. HEMO	OBEMO vs. FEMO
$f_1$	0.3521	1.21E-04	1.02E-04
$f_2$	0.4237	1.05E-04	0.88E-04
$f_3$	0.2189	4.84E-05	3.12E-05
$f_4$	0.4321	1.35E-05	1.09E-05
$f_5$	0.5281	2.73E-04	2.21E-04
$f_6$	0.4219	1.07E04	0.77E-04
$f_7$	0.3281	3.12E-05	2.45E-05
$f_8$	0.4209	4.01E-05	3.62E-05
$f_9$	0.2135	1.86E-05	1.29E-05

**Tabla 3.4:** Resultados de la prueba rangos de Wilcoxon considerando el “promedio del mejor valor de  $f(\mathbf{x})$ ”.

Función	Valores $p$		
	OBEMO vs. EMO	OBEMO vs. HEMO	OBEMO vs. FEMO
$f_1$	2.97E-04	0.2122	0.2877
$f_2$	3.39E-04	0.1802	0.2298
$f_3$	8.64E-07	1.1222	0.1567
$f_4$	7.54E-05	0.2183	0.1988
$f_5$	1.70E-04	0.3712	0.3319
$f_6$	5.40E-13	0.4129	0.3831
$f_7$	7.56E-04	0.3211	0.3565
$f_8$	1.97E-04	0.2997	0.2586
$f_9$	1.34E-05	0.3521	0.4011

**Tabla 3.5:** Resultados de la prueba rangos de Wilcoxon considerando el “promedio de iteraciones *MAXITER*”.

En la Tabla 3.4, los valores  $p$  para el caso de OBEMO vs. EMO son mayores que 0.05 (5% del nivel de significancia) lo cual es una fuerte evidencia de lo que menciona la hipótesis nula, por lo tanto no existe una diferencia significativa entre ambos métodos. Por otra parte, en los caso de OBEMO vs. HEMO y OBEMO vs. FEMO el valor de  $p$  es menor que 0.05, por lo tanto existe una diferencia significativa entre el promedio de los mejores valores de ambos métodos. Para obtener los datos de la Tabla 3.5 se aplicó el mismo criterio que se empleo para las analizar los valores de la función objetivo, es evidente que existe una diferencia signficante en el caso de OBEMO vs. EMO, contrario a los casos de OBEMO vs. HEMO y OBEMO vs. FEMO donde los resultados son similares.

*Funciones de prueba multidimensionales:* En contraste con las funciones originales, las funciones multidimensionales exhiben varios mínimos/máximos, los cuales son en general más difíciles de encontrar. De forma similar a las funciones originales, el rendimiento del algoritmo OBEMO es comparado con el de EMO, HEMO y FEMO. Esta comparación refleja la habilidad de los algoritmos para escapar de óptimos locales y localizar el óptimo global más cercano, en el menor número de iteraciones. Todas las funciones de prueba son de dimensión 30. Los resultados mostrados en la Tabla 3.6 son



promediados sobre 35 ejecuciones reportando el “promedio del mejor valor de  $f(\mathbf{x})$ ” y “promedio del número máximo de iteraciones o *MAXITER*” como índices de rendimiento. La Figura 3.6 muestra el proceso sobre dos de las funciones multidimensionales, tales valores corresponden al mejor caso de cada enfoque, obtenido después de 35 ejecuciones.

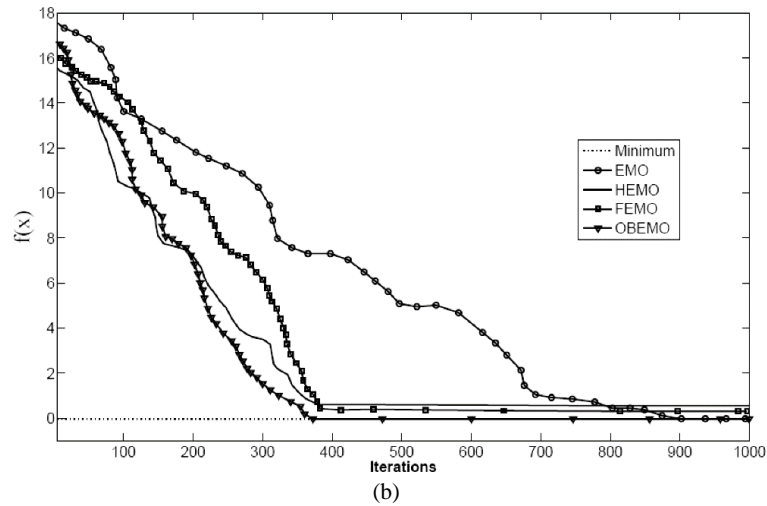
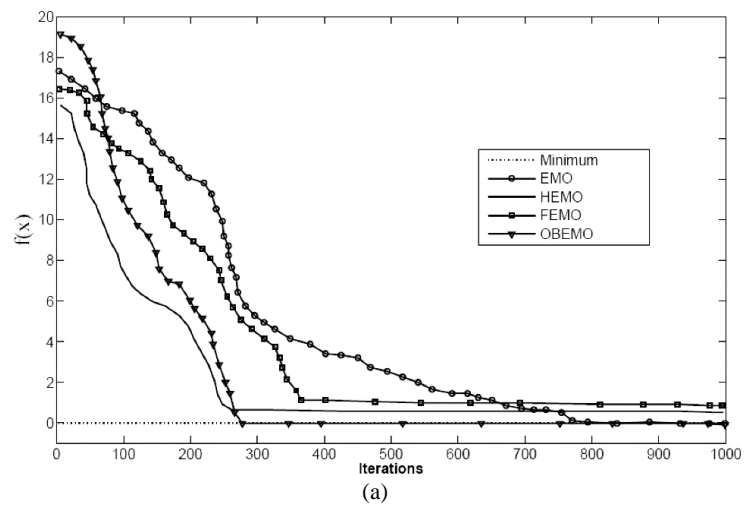
Los resultados de la prueba de rangos de Wilcoxon se presentan en la Tabla 3.7, que muestra los valores  $p$  que analizan el promedio del mejor valor en la función objetivo. Para el caso de OBEMO vs. EMO no indican que exista una diferencia significativa entre ambos métodos. Por su parte, los valores  $p$  para OBEMO vs. HEMO y OBEMO vs. FEMO prueban la evidencia de que existe una diferencia significativa entre los métodos. Por último, en la Tabla 3.8 se considera el análisis de Wilcoxon con respecto al promedio del número de iteraciones. Como se puede apreciar los valores  $p$  son similares a los resultados de las funciones originales. En la Figura 3.6 se muestra la evolución de la función objetivo para dos de problemas multidimensionales. Para esto son aplicados los diferentes algoritmos basados en EMO, en las gráficas se aprecia que el algoritmo OBEMO converge en menos iteraciones y con mayor precisión, en comparación con los otros métodos.

Función		$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$
Dimensión		30	30	30	30	30
EMO	Promedio del mejor valor $f(x)$	2.12E-05	1.21E-06	1.87E-05	1.97E-05	2.11E-06
	Promedio de <i>MAXITER</i>	622	789	754	802	833
OBEMO	Promedio del mejor valor $f(x)$	3.76E-05	5.88E-06	3.31E-05	4.63E-05	3.331E-06
	Promedio de <i>MAXITER</i>	222	321	279	321	342
HEMO	Promedio del mejor valor $f(x)$	2.47E-02	1.05E-02	2.77E-02	3.08E-02	1.88E-02
	Promedio de <i>MAXITER</i>	210	309	263	307	328
FEMO	Promedio del mejor valor $f(x)$	1.36E-02	2.62E-02	1.93E-02	2.75E-02	2.33E-02
	Promedio de <i>MAXITER</i>	241	361	294	318	353

**Tabla 3.6:** Resultados comparativos para los algoritmos EMO, OBEMO, HEMO y FEMO considerando las funciones multidimensionales (Tabla 3.2).

Función	Valores $p$		
	OBEMO vs. EMO	OBEMO vs. HEMO	OBEMO vs. FEMO
$f_{10}$	0.2132	3.21E-05	3.14E-05
$f_{11}$	0.3161	2.39E-05	2.77E-05
$f_{12}$	0.4192	5.11E-05	1.23E-05
$f_{13}$	0.3328	3.33E-05	3.21E-05
$f_{14}$	0.4210	4.61E-05	1.88E-05

**Tabla 3.7:** Resultados de la prueba rangos de Wilcoxon considerando el “promedio del mejor valor de  $f(\mathbf{x})$ ” de las funciones multidimensionales.



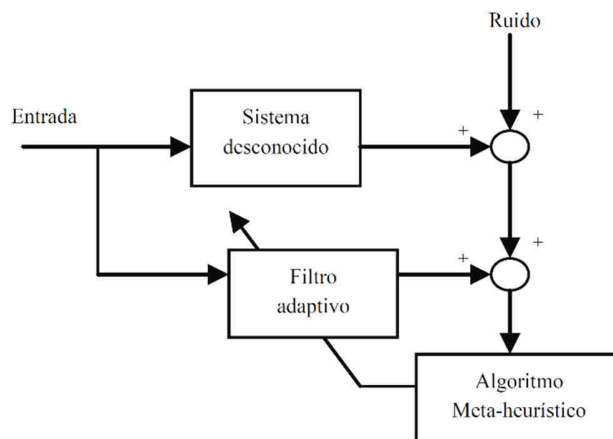
**Figura 3.6:** Comparación del proceso del optimización para dos de las funciones de prueba multidimensionales: (a)  $f_{12}$  y (b)  $f_{14}$ .

Función	Valores $p$		
	OBEMO vs. EMO	OBEMO vs. HEMO	OBEMO vs. FEMO
$f_{10}$	3.78E-05	0.1322	0.2356
$f_{11}$	2.55E-05	0.2461	0.1492
$f_{12}$	6.72E-05	0.3351	0.3147
$f_{13}$	4.27E-05	0.2792	0.2735
$f_{14}$	3.45E-05	0.3248	0.3811

**Tabla 3.8:** Resultados de la prueba rangos de Wilcoxon considerando el “promedio de iteraciones *MAXITER*” de las funciones multidimensionales.

### 3.4 Modelado de filtros digitales

En el Capítulo 2, se introdujo la teoría referente al diseño de filtros digitales, se definieron las ecuaciones típicas que modelan los diferentes tipos de filtros: respuesta al impulso finita (FIR) y respuesta al impulso infinita (IIR). Partiendo de esto se puede decir que, el proceso de identificación de un filtro consiste en modificar sus coeficientes, hasta que se asemeje a la función de transferencia de un sistema desconocido (Figura 3.7). Dadas estas características, es posible definir el modelado de filtros como un problema de optimización, en el cual se busca un conjunto de soluciones, que den como resultado un mínimo error entre la salida del filtro y la salida del sistema desconocido. Para dar solución a esto, es posible hacer uso de las técnicas de optimización meta-heurísticas.



**Figura 3.7:** Proceso de identificación de los parámetros de un filtro digital usando algoritmos metaheurísticos.

### 3.4.1 Filtros FIR e IIR

Dentro del procesamiento digital de señales los filtros se representan por medio de una función de transferencia. De esta forma, los filtros FIR se definen de forma general por la ecuación (3.9)

$$\frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_r z^{-r} \quad (3.9)$$

donde  $b = \{b_0, b_1, b_2 \dots b_r\}$  es un vector  $r$ -dimensional que contiene los parámetros del numerador, los valores de  $z^k$  corresponden al número de retardos que tiene la señal de entrada para  $k = -1, -2, \dots -r$ . Del mismo modo,  $Y(z)$  corresponde a la respuesta o salida del filtro, mientras que  $X(z)$  es la señal de entrada del filtro FIR. Por otra parte, un filtro IIR se define por la siguiente función de transferencia:

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_r z^{-r}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_c z^{-c}} \quad (3.10)$$

donde  $Y(z)$  y  $X(z)$  son la salida y la entrada del filtro, respectivamente. Entonces el vector de parámetros está representado por  $\mathbf{A} \cup \mathbf{B} = \{b_0, b_1, b_2, \dots, b_r, a_0, a_1, a_2, \dots, a_c\}$ , de tal forma que sus elementos se encuentran en las dimensiones  $\{r+c\}$ . Estos tipos de filtros son los más usados dentro del procesamiento digital de señales. El problema en el diseño de estos filtros se resume en la búsqueda de los valores de los vectores  $\mathbf{B}$  y  $\mathbf{A}$ , que mejor se aproximen a la salida del filtro en comparación con la salida de un modelo dado.

## 3.5 Modelado de filtros digitales usando el algoritmo EMO

En base a lo explicado en la sección 3.4 y en el Capítulo 2, se propone el uso del algoritmo *Electromagnetism-Like Optimization* para la identificación de los parámetros que modelen los filtros IIR. Primero es necesario definir cómo se representan las partículas dentro del enfoque EMO, después se define la función objetivo a optimizar y por último se implementa el algoritmo. A continuación se detalla cada uno de estos puntos.

### 3.5.1 Representación de las partículas

Bajo el enfoque EMO, las partículas son generadas de forma aleatoria dentro de un espacio acotado  $d$ -dimensional. Para la identificación de filtros IIR, la dimensión es

$d = r + c$ , que corresponde a la longitud de los vectores del numerador y denominador de dicho filtro. De esta manera, las partículas son almacenadas en un vector de soluciones  $\mathbf{S}$ , el cual contendrá  $m$  partículas de dimensión  $d$  cada una. De manera tal que  $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \dots, \mathbf{s}_m\}$  contiene las partículas de la forma  $\mathbf{s}_m = \{s_m^1, s_m^2, s_m^3, \dots, s_m^d\}^T$ , donde  $T$  se refiere al vector  $\mathbf{s}_m$  transpuesto.

### 3.5.2 Función objetivo

La calidad de dichas partículas se evalúa con respecto a la función objetivo, que mide la afinidad de las respuestas entre el sistema desconocido y el filtro estimado. Al evaluar cada partícula en el filtro digital IIR, éste genera una salida  $Y_{fil}$  la cual debe ser comparada con la salida del sistema desconocido  $Y_p$ . Esta diferencia es medida empleando el error cuadrático medio (*Mean Squared Error*, MSE) [55]. Este valor se considera como la función objetivo  $J$ , la cual se define formalmente como:

$$J(\mathbf{s}) = \frac{1}{Ti} \sum_{t=1}^{Ti} (Y_p(t) - Y_{fil}(t))^2 \quad (3.11)$$

Para la medición del MSE, tanto el sistema como el filtro son simulados durante un número de iteraciones (tiempo)  $Ti$ . Con fines de comparación, este parámetro se establece como  $Ti = 100$ , en base al trabajo propuesto por Howell y Gordon [55] y que a su vez es empleado ampliamente en la literatura referente. El valor de  $J$  debe ser minimizado por el algoritmo EMO, tras cada iteración los valores serán simulados y un nuevo valor de afinidad será asignado a cada partícula. Finalmente, tras un número definido de iteraciones o un valor  $J$  permisible (convergencia) se encuentran los parámetros óptimos del filtro.

### 3.5.3 Implementación

La implementación del algoritmo propuesto para la identificación y modelado de filtros IIR usando el algoritmo EMO se resume en los siguientes pasos:

- Paso 1:** Se generan  $m$  partículas iniciales aleatoriamente de acuerdo al Algoritmo 3.1. Cada partícula  $\mathbf{s}_i$  es evaluada en la función objetivo  $J$ . Se elige la mejor partícula  $\mathbf{s}^{mejor}$ , la cual es la que tiene el menor valor de  $J$ .
- Paso 2:** Se aplica la búsqueda local descrita en el Algoritmo 3.2 a todos los elementos de la población  $\mathbf{S}$ .

- Paso 3:** La carga entre las partículas se obtiene usando la ecuación (3.2), y su vector de fuerza se calcula con la ecuación (3.3). La partícula  $s^{mejor}$ , con un mejor valor en la función objetivo, mantiene una carga superior, y por lo tanto, una mayor fuerza de atracción o repulsión (Algoritmo 3.3).
- Paso 4:** Las partículas son desplazadas de acuerdo a la magnitud de su fuerza. La nueva posición de la partícula es calculada por la ecuación (3.5).  $s^{mejor}$  no es desplazada porque tiene la fuerza mayor y ésta atrae el resto de partículas hacia ella. (Algoritmo 3.4).
- Paso 5:** El índice de iteraciones del algoritmo EMO se incrementa. Se verifica el criterio de parada, si se ha alcanzado el máximo número de iteraciones o si el valor de  $J$  es menor o igual a un valor preestablecido, el flujo continúa al paso 6, de lo contrario se regresa al paso 3.
- Paso 6:** La mejor partícula  $s^{mejor}$  es seleccionada como la solución global de la población modificada en la última iteración.

Es importante señalar que cada vez que se evalúa la función objetivo, ésta recibe los valores de los parámetros del filtro y como se mencionó anteriormente son simulados por un periodo de tiempo prefijado para después extraer el MSE. De esta forma el valor de  $J$  que cada partícula posee, provee la información necesaria para determinar si una partícula se modifica o no.

#### 3.5.4 Resultados experimentales del algoritmo para identificación de parámetros y diseño de filtros IIR usando EMO

Para evaluar el desempeño del algoritmo propuesto en la identificación de filtros IIR se empleó un conjunto de problemas propuestos originalmente por Johnson y Larimor [177], y que son utilizados como pruebas estándar para calificar la calidad de identificación de una determinada propuesta. En general, estos problemas incluyen la identificación de sistemas, considerando la utilización de filtros IIR de diferente orden. Con el objetivo de verificar la eficiencia del algoritmo propuesto, los resultados se compararon con el enfoque de identificación propuesto en [57], basado en el algoritmo de optimización metaheurístico de optimización por enjambre de partículas (PSO). La descripción del trabajo experimental se presenta en el siguiente orden:

- Planta de segundo orden con un modelo de primer orden.
- Planta de segundo orden con un modelo de segundo orden.
- Planta de tercer orden con un modelo de segundo orden.
- Planta y filtro de orden superior.

- Comparaciones con el método basado en PSO.

Cada experimento realizado fue llevado a cabo con un conjunto de 10 partículas, un máximo de iteración para la búsqueda local de  $LSITER = 4$ , la longitud de paso para la búsqueda local es  $\delta = 0.025$ , y un valor máximo de iteraciones  $MAXITER = 3000$ . El número de iteraciones  $MAXITER$  fue elegido, con fines comparativos, a partir de otros trabajos similares [52, 55, 57].

*Planta de segundo orden con un modelo de primer orden:* Para este problema, la función de transferencia que define la planta a modelar, está dada por la siguiente ecuación:

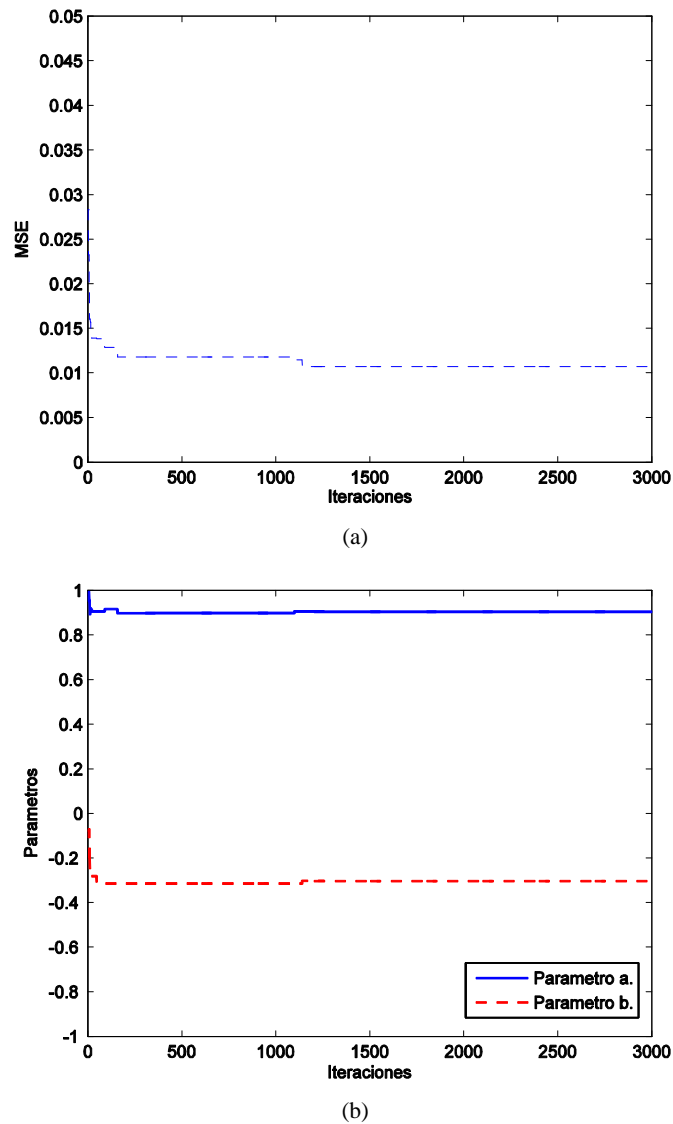
$$H_p(z^{-1}) = \frac{y_p(z^{-1})}{u(z^{-1})} = \frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} - 0.25z^{-2}} \quad (3.12)$$

dicha planta se pretende modelar por un filtro IIR con la siguiente función de transferencia.

$$H_m(z^{-1}) = \frac{y_m(z^{-1})}{u(z^{-1})} = \frac{b}{1 - az^{-1}} \quad (3.13)$$

Donde  $H_p(z^{-1})$  y  $H_m(z^{-1})$  corresponden a la relación entrada-salida de la planta y del modelo propuesto, respectivamente;  $u(z^{-1})$  es la entrada, la cual corresponde a una señal de ruido blanco, que es aplicada tanto a la planta como al modelo. Los valores  $a$  y  $b$  son modificados por el algoritmo EMO, este proceso se realiza haciendo las modificaciones pertinentes en base a los valores que toma la función objetivo  $J$ . Para este problema, el espacio de búsqueda se encuentra definido por los límites superior  $u = [1, 1]$  e inferior  $l = [-1, -1]$ .

La Figura 3.8 (a) muestra la evolución del MSE a lo largo de las 3000 iteraciones, se puede apreciar que dentro de las primeras 500 iteraciones el algoritmo logra converger a un valor cercano al óptimo. Por otra parte, la manera en que los valores  $a$  y  $b$  van evolucionando a lo largo de las iteraciones se muestra en la Figura 3.8 (b), donde se percibe que de forma similar al MSE, ambos parámetros logran establecer sus valores dentro de las primeras 500 iteraciones y después sólo se realizan ajustes mínimos para llegar a la convergencia.



**Figura 3.8:** (a) evolución del error cuadrático medio y (b) parámetros  $a$  y  $b$  a lo largo de las iteraciones, en ambos casos para la planta de segundo orden con un modelo de primer orden.

Los valores finales que calcula el algoritmo EMO, comparados con los que dan solución al problema, se presentan en la Tabla 3.9.

Valores	$a$	$b$	$J$
EMO	0.9034	-0.3030	0.0107
Reales	0.906	-0.311	-

**Tabla 3.9:** Valores encontrados por el algoritmo EMO y valores reales del modelo.



*Planta de segundo orden con un modelo de segundo orden:* para este problema, la planta se define por la siguiente función de transferencia:

$$H_p(z^{-1}) = \frac{y_p(z^{-1})}{u(z^{-1})} = \frac{1}{(1-0.7z^{-1})^2} = \frac{1}{1-1.4z^{-1}+0.49z^{-2}} \quad (3.14)$$

mientras que la ecuación (3.15) muestra la función de transferencia del filtro IIR propuesto.

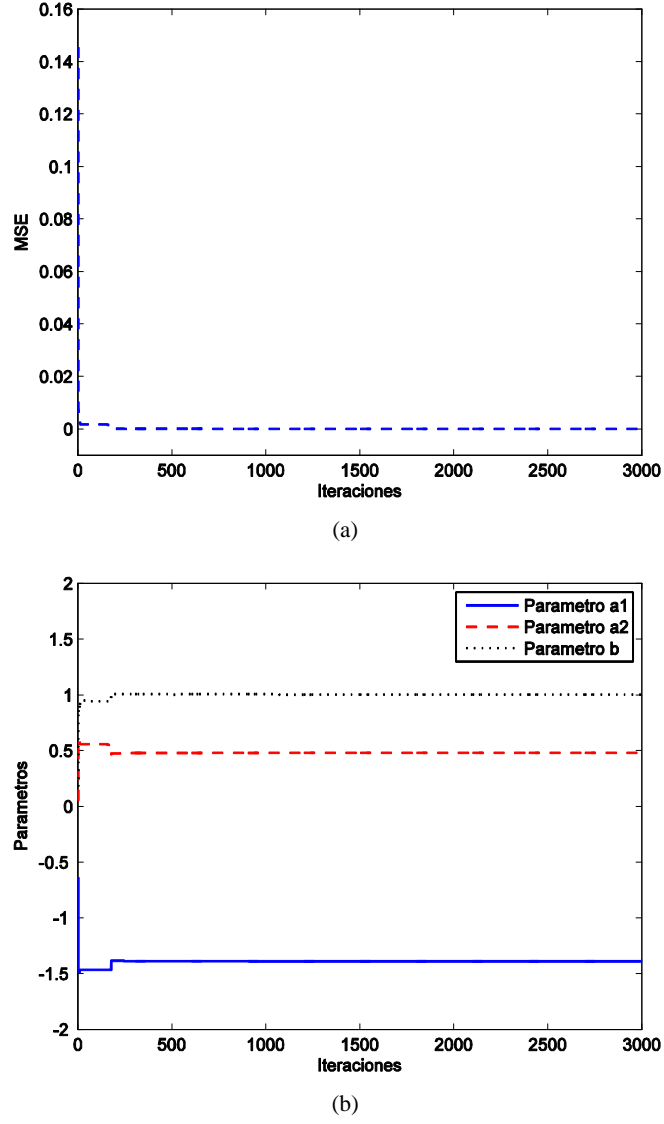
$$H_m(z^{-1}) = \frac{y_m(z^{-1})}{u(z^{-1})} = \frac{b}{1-a_1z^{-1}+a_20.49z^{-2}}, \quad (3.15)$$

$$H_{fir}(z^{-1}) = (1-0.7z^{-1})^2 \cdot (1+0.7z^{-1})^2$$

En el modelo, los parámetros a calibrar son  $a_1$ ,  $a_2$  y  $b$ , dando como resultado un problema de optimización de 3 dimensiones. La entrada  $u(z^{-1})$  en este caso, corresponde a una señal de ruido “coloreado”, el cual se genera a partir del filtro FIR que se muestra en la ecuación (3.15). Los límites de cada una de las dimensiones son:  $u = [2,1,1]$  y  $l = [-2,-1,-1]$ .

La Figura 3.9 (a) muestra cómo el MSE se minimiza drásticamente dentro de las primeras 100 iteraciones, finalmente se estaciona en un valor cercano a cero. La evolución de los tres parámetros durante el proceso de identificación se muestra en la Figura 3.9 (b). Estos parámetros toman diversos valores iniciales que son modificados en base a la función objetivo  $J$ , hasta que finalmente logran converger al valor que minimiza el MSE. La Tabla 3.10 presenta los valores finales de  $a_1$ ,  $a_2$  y  $b$  encontrados por el algoritmo EMO.

*Planta de tercer orden con un modelo de segundo orden:* el tercer problema propuesto por el conjunto de pruebas elegido [177], corresponde a una planta de tercer orden que se pretende modelar empleando un filtro IIR con una función de transferencia de segundo orden. La ecuación (3.16) define la estructura planta propuesta.



**Figura 3.9:** (a) evolución del error cuadrático medio y (b) parámetros  $a_1$ ,  $a_2$  y  $b$  a lo largo de las iteraciones, en ambos casos para la planta de segundo orden con un modelo de segundo orden.

Valores	$a_1$	$a_2$	$b$	$J$
EMO	-1.3936	0.4803	1.0005	2.463E-05
Reales	-1.4000	0.4900	1.0000	-

**Tabla 3.10:** Valores encontrados por el algoritmo EMO y valores reales del modelo.

$$H_p(z^{-1}) = \frac{y_p(z^{-1})}{u(z^{-1})} = \frac{1}{(1 - 0.6z^{-1})^3} \quad (3.16)$$

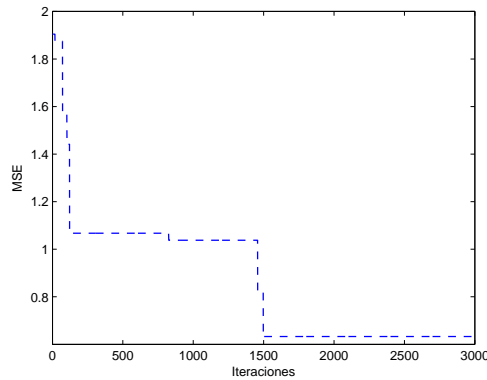
Por otra parte, en la ecuación (3.17) se propone el modelo IIR de segundo orden que aproxima la planta.

$$H_m(z^{-1}) = \frac{y_m(z^{-1})}{u(z^{-1})} = \frac{b}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (3.17)$$

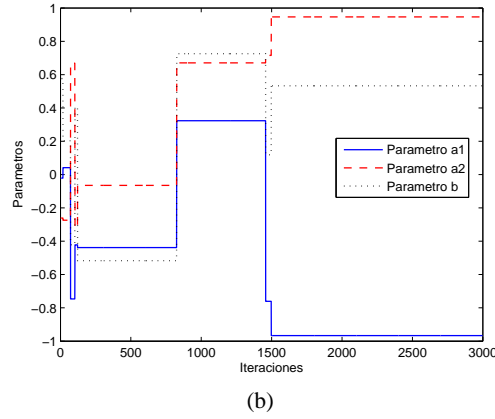
Al igual que en las sub-secciones anteriores, el problema de optimización se define como el problema de encontrar los valores de los parámetros  $a_1$ ,  $a_2$  y  $b$ , que minimicen el MSE existente entre las respuestas de  $y_p$  y  $y_m$ . Las cuales dependen de una señal de entrada  $u(z^{-1})$ , que contiene el ruido añadido por medio del filtro FIR, que se muestra a continuación:

$$H_{fir}(z^{-1}) = (1 - 0.6z^{-1})^2 \cdot (1 + 0.6z^{-1})^2 \quad (3.18)$$

El espacio de búsqueda de este problema consta de tres dimensiones, las cuales están acotadas por los siguientes límites superior  $u = [1, 1, 1]$  e inferior  $l = [-1, -1, -1]$ . En la Figura 3.10 (a) se muestra la gráfica del MSE a lo largo de 3000 iteraciones. Es importante mencionar que en este tipo de problemas el MSE no logra llegar a valores cercanos a cero, ya que el filtro con el que se está modelando no tiene el orden apropiado, por lo tanto, el algoritmo EMO trata de buscar los mejores valores que minimicen ese valor de error; sin embargo, esto no significa que los parámetros encontrados modelen con exactitud la planta deseada. En la Figura 3.10 (b) se puede ver cómo evolucionan los valores de los coeficientes. Se aprecia en la gráfica que aunque llegan a converger en un valor fijo, estos valores son los que corresponden al mínimo global de la superficie de error, sin que ello signifique que el modelo, debido a su bajo orden, identifique plenamente la planta.



(a)



**Figura 3.10:** (a) evolución del error cuadrático medio y (b) parámetros  $a_1$ ,  $a_2$  y  $b$  a lo largo de las iteraciones, en ambos casos para la planta de tercer orden con un modelo de segundo orden.

*Planta y filtro de orden superior:* hasta ahora, los filtros y las plantas propuestas han sido problemas en los cuales se han calibrado pocos parámetros. El algoritmo EMO ha arrojado resultados competitivos con esos problemas que pueden ser considerados como sencillos. Sin embargo, la mayor complicación en la identificación es encontrar los valores de los parámetros en filtros de orden mayor a tres. Para ejemplificar estos filtros se propone una planta de orden 6 en la ecuación (3.19), que debe ser modelada por un filtro IIR de orden 4 presentado en la ecuación (3.20).

$$H_p(z^{-1}) = \frac{y_p(z^{-1})}{u(z^{-1})} = \frac{1 - 0.4z^{-2} - 0.65z^{-4} + 0.26z^{-6}}{1 - 0.77z^{-2} - 0.8498z^{-4} + 0.6486z^{-6}} \quad (3.19)$$

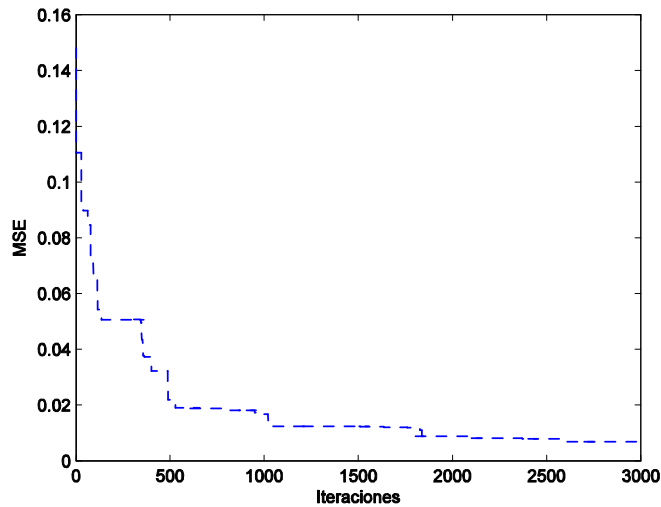
$$H_m(z^{-1}) = \frac{y_m(z^{-1})}{u(z^{-1})} = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4}}{1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-4}} \quad (3.20)$$

Los parámetros que deben calibrarse son  $a_1, a_2, a_3, a_4$ , asimismo  $b_0, b_1, b_2, b_3, b_4$ . Por lo tanto, como se explicó en la sección de identificación de filtros digitales usando el algoritmo EMO, el espacio de búsqueda es de nueve dimensiones y cada una tiene como límites superior  $u = 1.5$  e inferior  $l = -1.5$ . El algoritmo EMO presenta un buen rendimiento en espacios con grandes dimensiones [13], por tal motivo resulta idónea su aplicación para encontrar los múltiples valores que den solución a la identificación con filtros de orden superior.

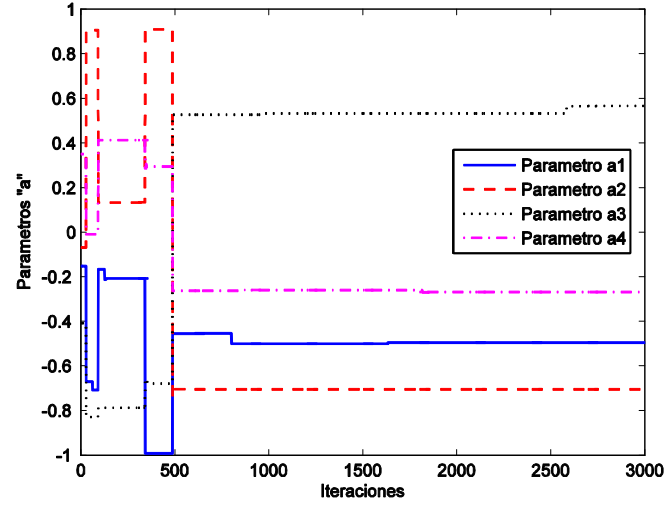
Los resultados experimentales muestran la eficiencia del algoritmo para minimizar el MSE. En la Figura 3.11 (a) se puede ver la gráfica de los valores que va tomando el error cuadrático medio durante las iteraciones. Este valor tiende a cero, hasta llegar a

estabilizarse. Es de esperar que al tratarse de un sistema de mayores dimensiones, el grado de dificultad sea también mayor. Por otra parte, la Figura 3.11 (b) presenta los valores que cada coeficiente IIR del denominador ( $a_1, a_2, a_3, a_4$ ) toma a lo largo de las iteraciones, mientras que la Figura 3.11 (c) muestra los valores de los parámetros  $b_0, b_1, b_2, b_3, b_4$ .

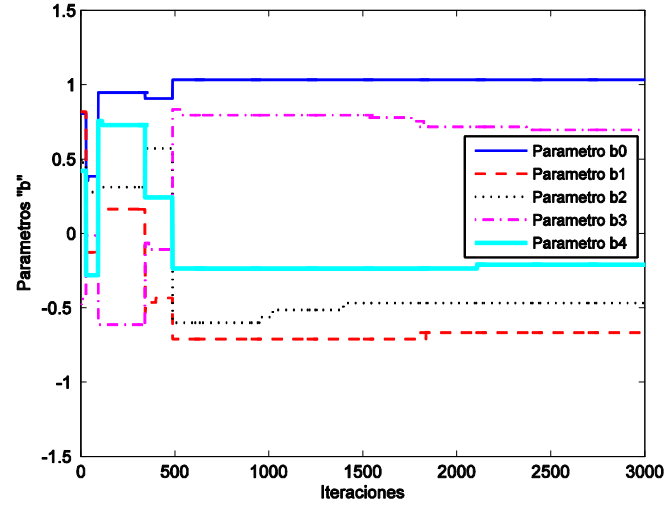
Dentro de las primeras 500 iteraciones el algoritmo basado en EMO realiza una exploración exhaustiva a lo largo del espacio de búsqueda hasta que se estabiliza o converge a los valores finales. Después, se realizan algunos ajustes finos sobre los valores de los parámetros. Ambos efectos corresponden respectivamente a la exploración y explotación del algoritmo EMO. Los parámetros del numerador ( $b_0, b_1, b_2, b_3, b_4$ ) a partir de sus valores iniciales aleatorios cambian abruptamente, pero después de transcurrir cierto número de iteraciones logran converger, para finalmente realizar los ajustes finales que minimicen el error de la manera más óptima posible. Los resultados que se presentan en las Figuras 3.11 (a), 3.11 (b) y 3.11 (c) exponen las capacidades del algoritmo EMO para la optimización multidimensional.



(a)



(b)



(c)

**Figura 3.11:** (a) evolución del error cuadrático medio y (b) parámetros  $a_1, a_2, a_3, a_4$  y (c) parámetros  $b_0, b_1, b_2, b_3, b_4$  a lo largo de las iteraciones, para un filtro de orden 4.

Los valores finales encontrados por el algoritmo después de las 3000 iteraciones se expresan en la Tabla 3.11, mientras el valor final del MSE es de 0.0069.

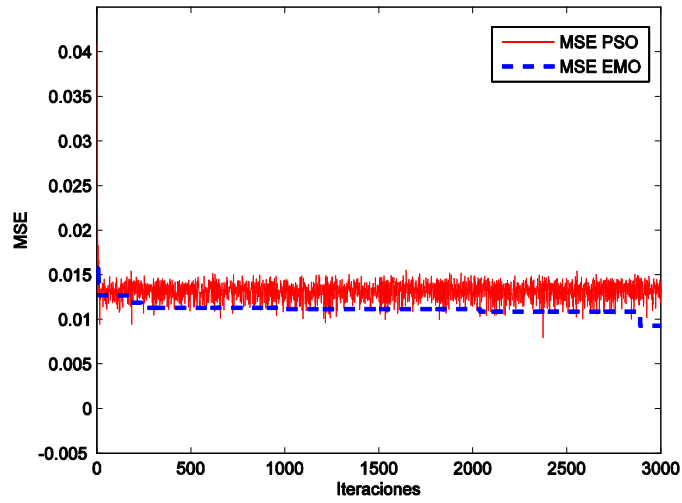
No. de parámetro	Valores $a$	No. de parámetro	Valores $b$
$a_1$	-0.4950	$b_0$	1.0335
$a_2$	-0.7049	$b_1$	-0.6670
$a_3$	0.5656	$b_2$	-0.4682
$a_4$	-0.2691	$b_3$	0.6961
-	-	$b_4$	-0.2100

**Tabla 3.11:** Valores encontrados por el algoritmo para los parámetros del numerado y los del denominador.

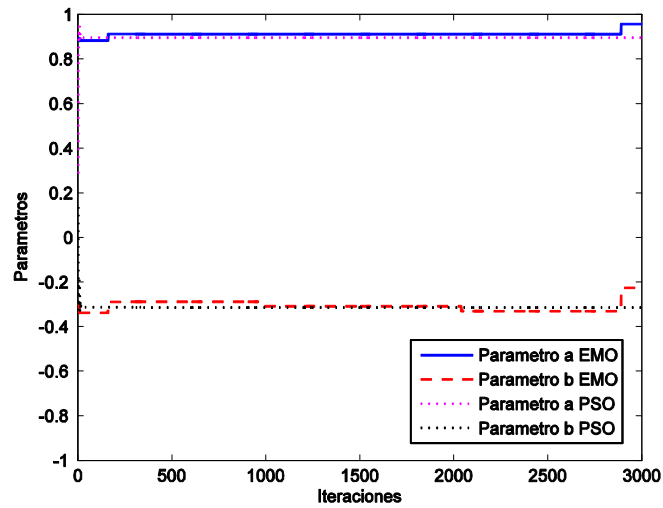
*Comparaciones con el método de identificación basado en PSO:* se ha realizado la comparación del método basado en EMO con el algoritmo de identificación propuesto por Lin et al. [57], basado en el popular método PSO [10]. La comparación se ha llevado a cabo con cada uno de los problemas analizados previamente: planta de segundo orden con un modelo de primer orden, planta de segundo orden con un modelo de segundo orden y planta y filtro de orden superior. Del mismo modo cada prueba se efectuó con 3000 iteraciones. Los resultados se describen a continuación.

La Figura 3.12 (a) muestra las gráficas del MSE, tanto del algoritmo basado en EMO, como del algoritmo PSO, para el caso del experimento descrito en la planta de segundo orden con un modelo de primer orden. Se puede ver claramente que ambos algoritmos minimizan el error en un número de iteraciones reducido. Sin embargo, el algoritmo PSO mantiene una serie de iteraciones a lo largo de un valor cercano a cero. Se observa igualmente que el algoritmo EMO posee un comportamiento estable, realizando sólo un pequeño ajuste para minimizar el valor. La Tabla 3.12 presenta los resultados de ambos algoritmos, asimismo, los valores óptimos reales que dan solución al problema. La evolución de los valores que toma cada uno de los parámetros durante las iteraciones realizadas se muestra en la Figura 3.12 (b).

Continuando con la comparación, se toma el segundo ejemplo, que consiste en una planta de segundo orden, modelada con un filtro IIR de segundo orden, para este caso la entrada corresponde a una señal a la que se le ha añadido ruido.



(a)



(b)

**Figura 3.12:** (a) comparación de los valores de MSE de los algoritmos EMO y PSO y (b) comparación de los parámetros  $a$  y  $b$  encontrados por EMO y PSO durante 30000 iteraciones. En ambos casos para una planta de segundo orden modelada con un filtro de primer orden.

Valores	$a$	$b$	$J$
EMO	0.9034	-0.3030	0.0107
PSO	0.8950	-0.3142	0.0121
Reales	0.906	-0.311	-

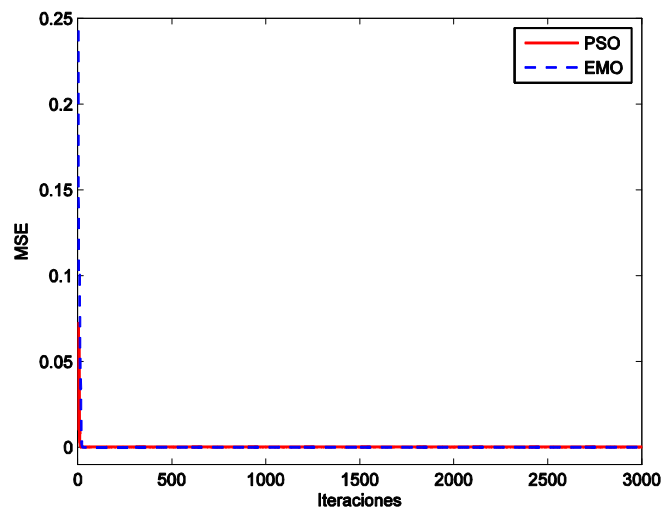
**Tabla 3.12:** Valores comparativos de los parámetros  $a$ ,  $b$ , y  $J$  encontrados tanto por el algoritmo EMO como el PSO.



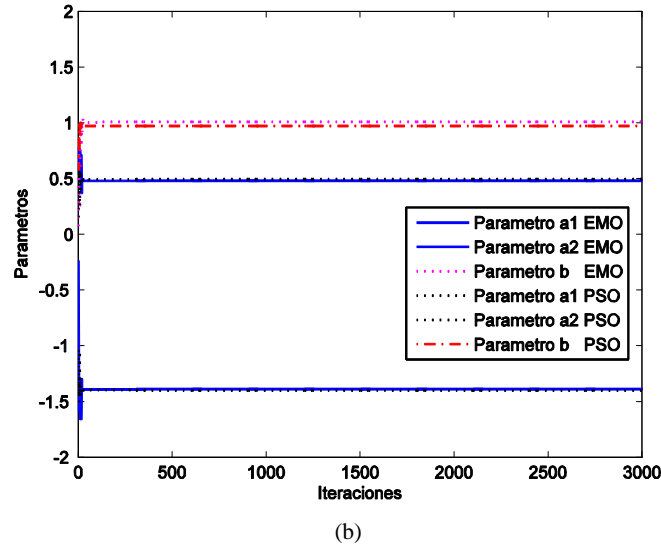
La Figura 3.13 (a) muestra los valores del MSE que ambos algoritmos (EMO y PSO) generaron durante las iteraciones. Para este problema, ambos algoritmos presentan un rendimiento similar, sin embargo, el algoritmo EMO logra converger en un menor número de iteraciones y mantener un valor MSE menor que el algoritmo PSO. La evolución de los valores de ambas propuestas se presenta en la Figura 3.13 (b), en la cual se aprecia que tanto el método basado en EMO como el PSO requieren un número reducido de iteraciones para encontrar los valores de los parámetros  $a_1$ ,  $a_2$  y  $b$ , con los cuales el MSE tiende a un valor cercano a cero. Cabe señalar, que a pesar de que las dos técnicas muestran comportamientos similares, el método propuesto es el que mejor aproxima los valores óptimos, tal como se puede apreciar en la Tabla 3.13.

Valores	$a_1$	$a_2$	$b$	$J$
EMO	-1.3936	0.4803	1.0005	2.463E-05
PSO	-1.4024	0.4925	0.9706	1.4113E-04
Reales	-1.4000	0.4900	1.0000	-

**Tabla 3.13:** Valores comparativos de los parámetros  $a$ ,  $b$ , y  $J$  encontrados tanto por el algoritmo EMO como el PSO.

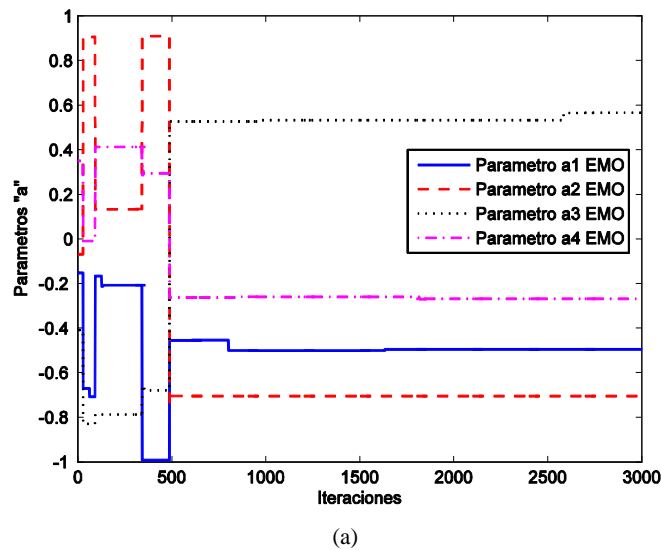


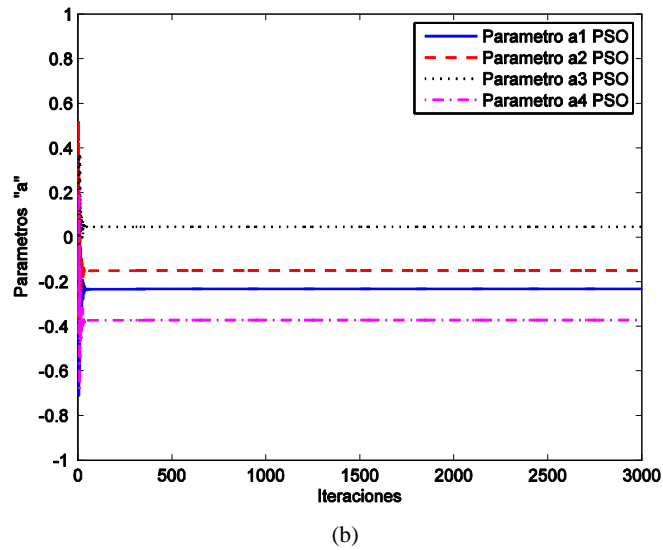
(a)



**Figura 3.13:** (a) comparación de los valores de MSE de los algoritmos EMO y PSO y (b) comparación de los parámetros  $a$  y  $b$  encontrados por EMO y PSO durante 30000 iteraciones. En ambos casos para una planta de segundo orden modelada con un filtro de segundo orden.

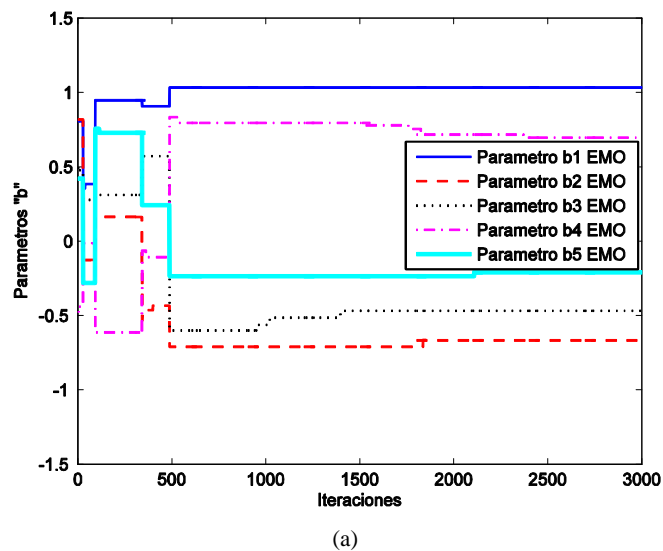
Finalmente se realizó la comparación de ambas técnicas de optimización con el problema que propone modelar una planta de orden 6 con un filtro IIR de orden 4. Del mismo modo que en las otras comparaciones, ambos algoritmos se ejecutan durante un número definido de iteraciones, durante las cuales se buscan los mejores valores que minimizan la función de error MSE. En este caso, el problema consta de 9 dimensiones. La Figura 3.14 presenta las evoluciones de cada parámetro del denominador del filtro IIR identificado, considerando ambos algoritmos.

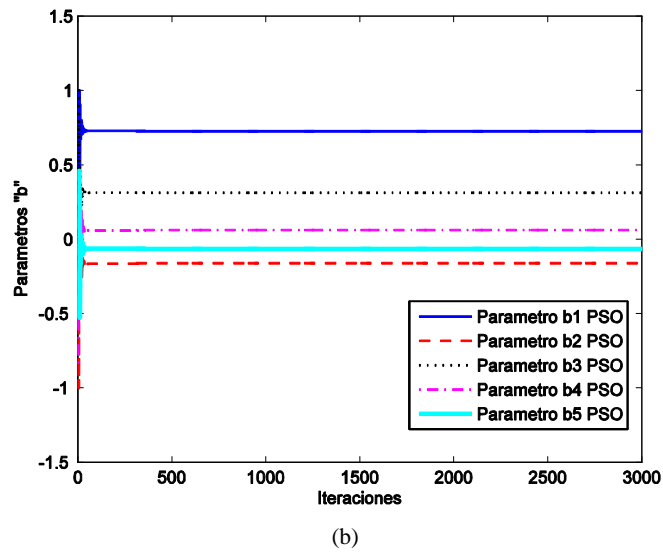




**Figura 3.14:** Graficas de los parámetros del denominador del filtro de orden 4 para los algoritmos (a) EMO y (b) PSO.

Por otro lado, los parámetros del numerador del filtro, que tanto el algoritmo EMO como el PSO identificaron durante las iteraciones, se muestran en la Figura 3.15. El error cuadrático medio de cada algoritmo se muestra en la Figura 3.16. Los valores encontrados en la última iteración se presentan en la Tabla 3.14.





**Figura 3.15:** Gráficas de los parámetros del numerador del filtro de orden 4 para los algoritmos (a) EMO y (b) PSO.

No. de parámetro	Valores EMO	Valores PSO	No. de parámetro	Valores EMO	Valores PSO
$a_1$	-0.4950	-0.2327	$b_0$	1.0335	0.7254
$a_2$	-0.7049	-0.1496	$b_1$	-0.6670	-0.1618
$a_3$	0.5656	-0.0469	$b_2$	-0.4682	0.3113
$a_4$	-0.2691	-0.3720	$b_3$	0.6961	0.0601
-	-	-	$b_4$	-0.2100	-0.0673

**Tabla 3.14:** Valores encontrados por el algoritmo EMO y PSO para los parámetros del numerador y los del denominador de un filtro de orden 4.

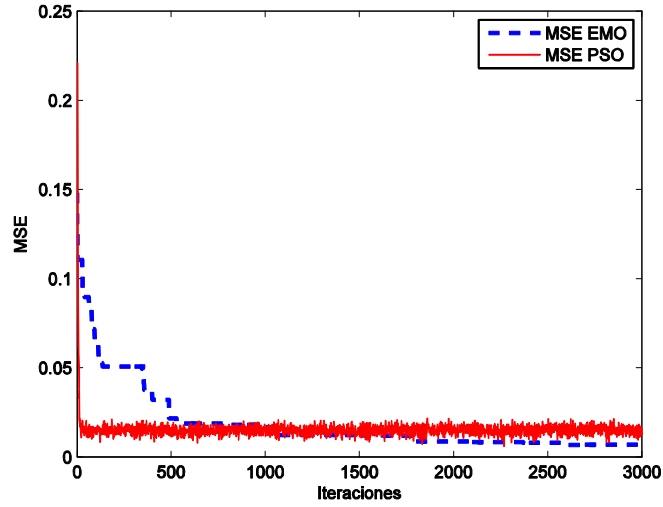


Figura 3.16: Comparación de los valores de MSE de los algoritmos EMO y PSO.

### 3.6 Detección de círculos en imágenes digitales usando el algoritmo EMO

Este apartado sirve como preámbulo para los procesos posteriores que se desarrollarán en el resto de secciones de este capítulo. Se presenta la implementación del algoritmo metaheurístico EMO para la detección de formas geométricas circulares en imágenes digitales. Con tal propósito, primero es necesario definir el manejo de los datos de la imagen, así como la representación de las partículas de la población. Un punto importante a considerar es la definición de la función objetivo, la cual permite abordar el problema desde el punto de vista de la optimización.

#### 3.6.1 Pre-procesamiento de los datos

Con el propósito de detectar formas circulares, las imágenes candidatas deben primero ser pre-procesadas por el bien conocido algoritmo de Canny [143, 144], el cual produce imágenes de bordes con un píxel de grosor. Entonces, las coordenadas  $(x_i, y_i)$  para cada píxel de bordes  $p_i$  son almacenadas dentro del vector de bordes  $\mathbf{P} = \{p_1, p_2, \dots, p_{NP}\}$ , con  $NP$  como el número total de píxeles de borde en la imagen.

### 3.6.2 Representación de las partículas

Para construir cada partícula o círculo candidato denominado  $\mathbf{C}$ , los índices  $e_1$ ,  $e_2$  y  $e_3$ , los cuales representan tres puntos de borde, previamente almacenados en el vector  $\mathbf{P}$ , deben agruparse asumiendo que están conectados mediante el contorno o circunferencia. Por lo tanto, el círculo  $\mathbf{C} = \{p_{e_1}, p_{e_2}, p_{e_3}\}$  que pase por todos los puntos puede ser considerado como una solución potencial al problema de detección. Considerando la configuración de los puntos de borde mostrados en la Figura 3.17, el centro del círculo  $\mathbf{C}$  con las coordenadas  $(x_0, y_0)$  y el radio  $r$  pueden caracterizarse como sigue:

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \quad (3.21)$$

donde  $x_0$  y  $y_0$  se calculan a través de las siguientes ecuaciones:

$$\begin{aligned} x_0 &= \frac{\det(\mathbf{A})}{4((x_{e_2} - x_{e_1})(y_{e_3} - y_{e_1}) - (x_{e_3} - x_{e_1})(y_{e_2} - y_{e_1}))}, \\ y_0 &= \frac{\det(\mathbf{B})}{4((x_{e_2} - x_{e_1})(y_{e_3} - y_{e_1}) - (x_{e_3} - x_{e_1})(y_{e_2} - y_{e_1}))} \end{aligned} \quad (3.22)$$

con  $\det(\mathbf{A})$  y  $\det(\mathbf{B})$  representando los determinantes de las matrices  $\mathbf{A}$  y  $\mathbf{B}$ , respectivamente, las cuales se definen como:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} x_{e_2}^2 + y_{e_2}^2 - (x_{e_1}^2 + y_{e_1}^2) & 2 \cdot (y_{e_1} - y_{e_1}) \\ x_{e_3}^2 + y_{e_3}^2 - (x_{e_1}^2 + y_{e_1}^2) & 2 \cdot (y_{e_3} - y_{e_1}) \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} 2 \cdot (x_{e_2} - x_{e_1}) & x_{e_2}^2 + y_{e_2}^2 - (x_{e_1}^2 + y_{e_1}^2) \\ 2 \cdot (x_{e_3} - x_{e_1}) & x_{e_3}^2 + y_{e_3}^2 - (x_{e_1}^2 + y_{e_1}^2) \end{bmatrix} \end{aligned} \quad (3.23)$$

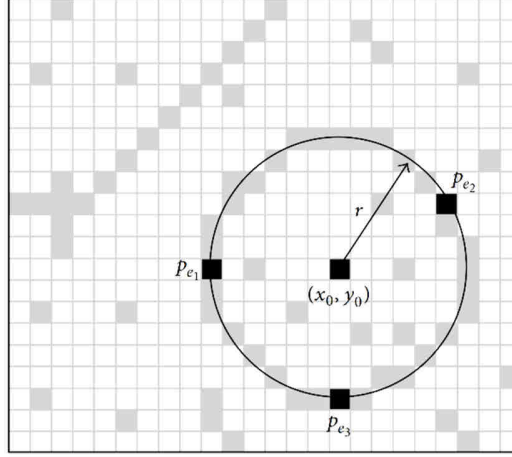
el radio  $r$  por lo tanto puede ser calculando usando:

$$r = \sqrt{(x_0 - x_{e_d})^2 + (y_0 - y_{e_d})^2} \quad (3.24)$$

donde  $d \in \{1, 2, 3\}$  y  $(x_{e_d}, y_{e_d})$  son las coordenadas de alguno de los tres puntos seleccionados que definen la partícula (o círculo)  $\mathbf{C}$ . La Figura 3.17 ilustra los parámetros principales definidos por las ecuaciones (3.21-3.24). Por lo tanto, los parámetros que conforman el círculo  $\{x_0, y_0, r\}$  pueden ser representados como una transformación  $T$  de los índices del vector de bordes  $e_1$ ,  $e_2$  y  $e_3$  como se muestra a continuación:

$$[x_0, y_0, r] = T(e_1, e_2, e_3) \quad (3.25)$$

Mediante la exploración de cada índice como una partícula, es posible hacer un barrido del espacio continuo, mientras se realiza la búsqueda de los parámetros para la forma geométrica, por medio del algoritmo EMO. Este enfoque reduce el espacio de búsqueda mediante la eliminación de las soluciones no factibles.



**Figura 3.17:** Círculo candidato (partícula cargada) construido a partir de la combinación de los puntos de borde  $p_{e1}$ ,  $p_{e2}$  y  $p_{e3}$ .

### 3.6.3 Función objetivo

Con el propósito de modelar la función de afinidad, las coordenadas de la circunferencia que define el círculo candidato  $\mathbf{C}$  son calculadas como una forma geométrica “virtual”, la cual debe validarse. Esto significa que se tiene que verificar si  $\mathbf{C}$  realmente existe en la imagen. Las coordenadas de los píxeles que componen la circunferencia se agrupan en el conjunto de prueba  $\mathbf{H} = \{h_1, h_2, \dots, h_{Ns}\}$ ,  $Ns$  representa la cantidad de puntos sobre los cuales debe verificarse la existencia de puntos de borde que corresponden a  $\mathbf{C}$ .

El conjunto  $\mathbf{H}$  se genera usando el algoritmo de punto medio para círculos (*mid point circle algorithm*, MCA) [178], el cual es un método ampliamente conocido, para determinar los puntos que describen el contorno de un círculo. MCA requiere como entrada sólo el radio  $r$  y las coordenadas del centro  $(x_0, y_0)$ , considerando sólo el primer octante de la ecuación del círculo  $x^2 + y^2 = r^2$ . Esta ecuación dibuja una curva que se inicia en el punto  $(r, 0)$  y procede hacia arriba a la izquierda mediante el uso de adiciones y sustracciones de números enteros. El propósito del MCA es calcular los puntos que requiere  $\mathbf{H}$  para representar el círculo candidato. A pesar de que esta técnica es considerada como la

más rápida y que además provee precisión a nivel de píxel, es importante asegurar que los puntos obtenidos, a partir del plano de la imagen, no sean considerados en el vector  $\mathbf{H}$ .

La función objetivo  $J(\mathbf{C})$  representa el error de coincidencia producido entre los píxeles  $\mathbf{H}$  del círculo candidato  $\mathbf{C}$  también llamada partícula y los píxeles que actualmente existen en la imagen de bordes, produciendo la siguiente ecuación:

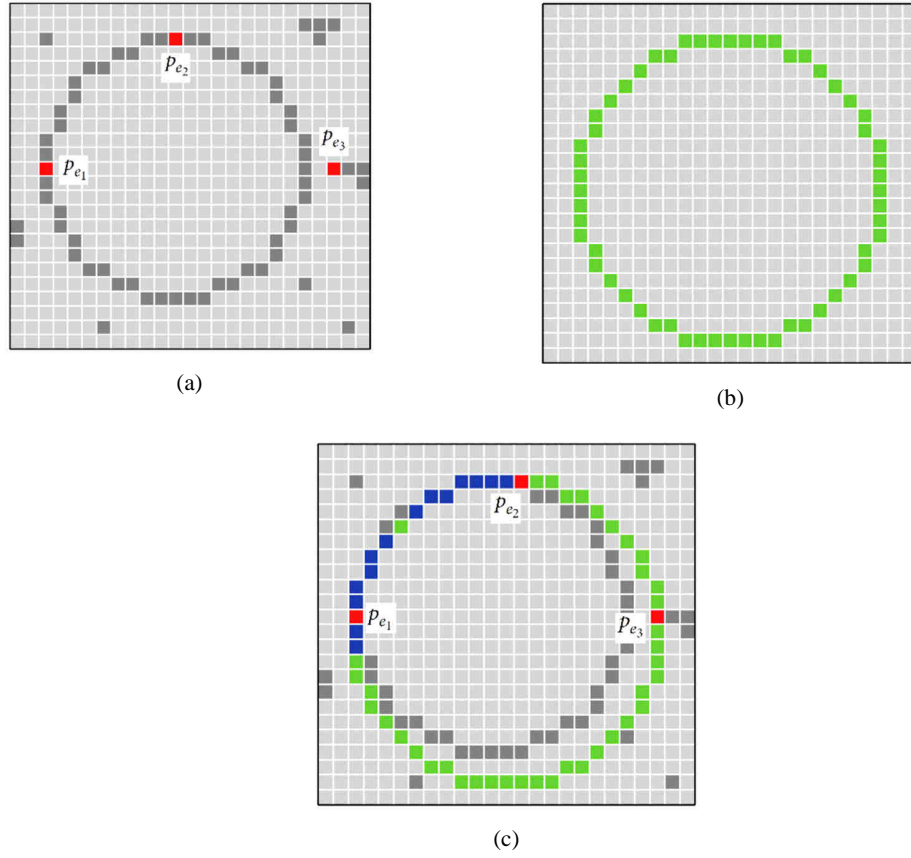
$$J(\mathbf{C}) = 1 - \frac{\sum_{v=1}^{Ns} E(h_v)}{Ns} \quad (3.26)$$

donde  $E(h_v)$  es una función que verifica la existencia del píxel en  $h_v$ ,  $h_v \in \mathbf{H}$  y  $Ns$  es la cantidad de elementos que posee  $\mathbf{H}$ . De esta forma  $E(h_v)$  se define como:

$$E(h_v) = \begin{cases} 1, & \text{si el píxel de prueba } h_v \text{ es un punto de borde} \\ 0, & \text{si no lo es} \end{cases} \quad (3.27)$$

Un valor de  $J(\mathbf{C})$  cercano a cero implica una mejor respuesta de “circularidad”. La Figura 3.18 muestra el procedimiento para evaluar una solución candidata  $\mathbf{C}$  con su respectiva representación como una forma geométrica virtual  $\mathbf{H}$ . La Figura 3.18 (a) presenta el mapa original de bordes, mientras que la Figura 3.18 (b) representa a  $\mathbf{H}$  que es el círculo virtual de la partícula  $\mathbf{C} = \{p_{e_1}, p_{e_2}, p_{e_3}\}$ . En la Figura 3.18 (c),  $\mathbf{H}$  se compara con la imagen original, punto por punto, con el propósito de encontrar píxeles coincidentes. Como ejemplo de esto, el individuo  $p$  de la población, se ha construido a partir de los puntos  $p_{e_1}$ ,  $p_{e_2}$  y  $p_{e_3}$  los cuales se muestran en la Figura 3.18 (a). La forma virtual  $\mathbf{H}$  obtenida usando el MCA, consta de 56 puntos ( $Ns = 56$ ), pero sólo 18 de ellos existen en las imágenes, marcándose como puntos azules y rojos en la Figura 3.18 (c), por lo tanto la sumatoria de  $E(h_v) = 18$  y  $J(\mathbf{C}) \approx 0.67$ .





**Figura 3.18:** Procedimiento para evaluar la función objetivo. (a) es la imagen original, (b) muestra la forma virtual  $H$  que corresponde a  $C$  y (c) tiene las coincidencias entre ambas imágenes.

#### 3.6.4 Detección de círculos usando el algoritmo EMO

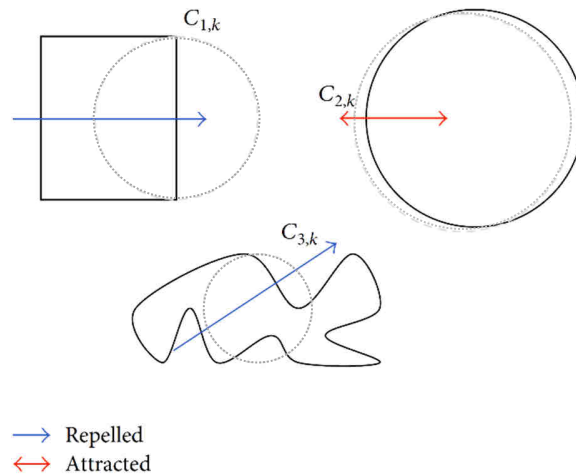
La implementación del algoritmo propuesto se puede resumir en los siguientes pasos:

- Paso 1:** Se aplica el filtro de Canny para encontrar los bordes, siendo éstos almacenados en el vector  $\mathbf{P} = \{p_1, p_2, \dots, p_{NP}\}$  y el índice  $iter$  toma el valor inicial de 1.
- Paso 2:** Se generan  $m$  partículas iniciales de forma aleatoria (Algoritmo 3.1), siendo  $\mathbf{C}_{a,1}$ ,  $a \in [1, m]$ . Las partículas con un radio demasiado pequeño o demasiado grande, son eliminadas y los puntos co-lineales descartados.
- Paso 3:** La función objetivo  $J(\mathbf{C}_{a,k})$  es evaluada para determinar cuál es la mejor partícula  $\mathbf{C}^B$ .
- Paso 4:** Se calcula la carga entre las partículas usando la ecuación (3.2) y el vector de fuerza mediante las ecuaciones (3.3 y 3.4). La partícula con el mejor valor en

la función objetivo mantiene una mejor carga y por consiguiente una mayor fuerza de atracción.

- Paso 5:** Las partículas son desplazadas de acuerdo a la magnitud de su fuerza (Algoritmo 3.4). La nueva posición de las partículas es  $C_a^y$ ,  $C^B$  no es desplazada porque debido a su fuerza atrae a las demás partículas hacia ella.
- Paso 6:** Para cada nueva posición  $C_a^y$  se aplica la búsqueda local de acuerdo al Algoritmo 3.2.
- Paso 7:** Se incrementa el índice *iter* y se verifica si se cumple el criterio de parada. Si tal criterio se cumple se va al paso 8, en caso contrario se regresa al paso 3.
- Paso 8:** La mejor partícula  $C^B$  se selecciona de la última iteración.
- Paso 9:** Los puntos correspondientes a  $C^B$  son eliminados del mapa de bordes original. Para el caso de detección de múltiples círculos, se regresa al paso 2.
- Paso 10:** Finalmente, la mejor partícula  $C^B$ , correspondiente a cada círculo detectado se usa para trazar sobre la imagen original el contorno de cada uno de ellos.

La Figura 3.19 muestra una analogía del método propuesto con la ley de Coulomb. Los objetos originales que deben ser detectados se representan por medio de una línea negra sólida, mientras que las formas con líneas discontinuas en color gris representan los círculos candidatos. Dado que los círculos candidatos  $C_{1,k}$  y  $C_{3,k}$  poseen un mayor valor en la función objetivo son repelidos (líneas azules), alejándose de las formas. En contraste, el círculo candidato  $C_{2,k}$  que posee un menor valor en la función objetivo es atraído (línea roja) hacia la forma circular contenida en la imagen.



**Figura 3.19:** Analogía con la ley de Coulomb.

### 3.7 Detector de glóbulos blancos usando el algoritmo EMO

Con el propósito de detectar glóbulos blancos en imágenes médicas, se propone un detector que combina el algoritmo para identificar círculos propuesto en la sección 3.6 con una nueva función objetivo.

#### 3.7.1 Pre-procesamiento de la imagen

Para poder aplicar el detector propuesto en las imágenes de frotis sanguíneo, éstas deben ser primero pre-procesadas para obtener dos nuevas imágenes: la imagen segmentada y su correspondiente mapa de bordes. La imagen segmentada es producida por medio de una estrategia de segmentación, mientras que el mapa de bordes es extraído por medio de alguno de los métodos más comunes para este fin. Ambas imágenes son consideradas por la nueva función objetivo para medir la similitud de un círculo candidato, con un glóbulo blanco de la imagen.

El objetivo de la estrategia de segmentación es aislar los glóbulos blancos de las otras estructuras como son: los glóbulos rojos y píxeles de fondo. La información de color, brillo y gradiente es comúnmente usada en los esquemas de umbralización, para generar las etiquetas que clasifican cada píxel. Aunque un algoritmo de umbralización simple puede ser usado para segmentar las células sanguíneas, para el método propuesto se emplea la técnica de expectación – maximización difusa (*diffused expectation – maximization*, DEM), la cual ha demostrado su eficiencia, al producir buenos resultados para esta aplicación [179].

DEM es un algoritmo basado en expectación – maximización (EM) el cual ha sido usado para segmentar imágenes médicas complejas, como se puede ver en [180]. En contraste con los algoritmos EM clásicos, el método DEM considera correlaciones espaciales entre los píxeles como parte del criterio de minimización. Tal adaptación permite segmentar objetos en presencia de ruido y condiciones complejas de la imagen.

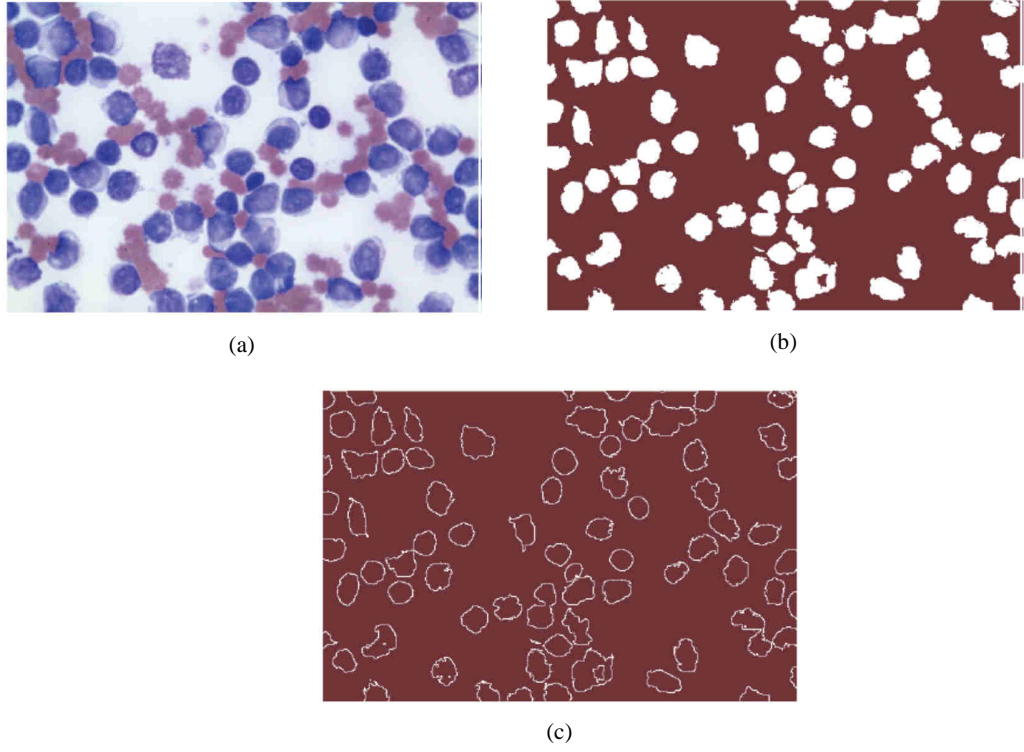
Para la segmentación de los glóbulos blancos, el algoritmo DEM ha sido configurado usando tres diferentes clases ( $K=3$ ), junto con los siguientes valores y parámetros  $g(\nabla h_{ik}) = |\nabla h_{ik}|^{-9/5}$ ,  $\lambda = 0.1$  y  $m = 10$  iteraciones. Estos valores se consideran la mejor configuración de acuerdo a [179]. Como resultado final de la aplicación de DEM, se obtienen tres umbrales: el primero corresponde a los glóbulos blancos, el segundo a los glóbulos rojos y el tercero el fondo de la imagen. La Figura 3.20 (b) presenta los resultados

de segmentación obtenidos por el enfoque DEM considerando como imagen de entrada la Figura 3.20 (a).

Una vez que se ha generado la imagen segmentada, se obtiene la imagen de bordes. El propósito de la imagen de bordes es obtener una representación simple que conserve las estructuras de los objetos contenidos en la imagen original. Los detectores de círculos basados en optimización [76, 78-81] operan directamente sobre el mapa de bordes con la finalidad de reconocer las formas circulares. Un gran número de algoritmos pueden usarse para extraer el mapa de bordes. Sin embargo, para el enfoque propuesto se usa la detección de bordes morfológica [145]. Este método de detección es uno de los clásicos en detección de bordes en imágenes binarias, en el cual las imágenes originales ( $I_B$ ) son erosionadas por medio de un elemento estructural simple ( $I_E$ ). Entonces, la imagen erosionada es invertida ( $\overline{I_E}$ ) y comparada con la imagen original ( $\overline{I_E} \wedge I_B$ ) con la finalidad de detectar los píxeles presentes en ambas imágenes. Tales píxeles conforman el mapa de bordes de  $I_B$ . La Figura 3.20 (c) muestra el mapa de bordes obtenido usando el procedimiento del detector de bordes morfológico.

### 3.7.2 El detector de círculos basado en EMO modificado

El enfoque de detección de círculos usa la codificación de tres puntos de borde que representan círculos candidatos en la imagen. En el detector de círculos basado en EMO propuesto en la sección 3.6, la calidad de cada individuo se calcula por medio de una función objetivo que evalúa la existencia de un círculo candidato considerando sólo la información del mapa de bordes, es decir, la forma de las estructuras de la imagen. Cuanto más se aproxime el círculo candidato al borde del círculo actual, más decrece el valor de la función objetivo. Por lo tanto, el desempeño de la detección depende de la calidad con que el mapa de bordes es obtenido a partir de la imagen original. Sin embargo, ya que las imágenes de frotis sanguíneo presentan diferentes condiciones y diferentes intensidades al ser teñidas, éstas producen mapas de bordes parcialmente dañados por píxeles con ruido. Bajo estas condiciones, el uso del detector de círculos basado en EMO no puede ser directamente implementado para identificar glóbulos blancos.



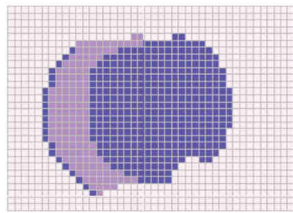
**Figura 3.20:** Pre-procesamiento: (a) imagen original de frotis sanguíneo, (b) imagen segmentada obtenida por DEM y (c) el mapa de bordes obtenido por medio del de detector de bordes morfológico.

Con el objetivo de usar el detector de círculos implementado con EMO bajo el contexto de la localización de células sanguíneas, es necesario cambiar la función objetivo presentada en la ecuación (3.26). En el enfoque propuesto, se ha creado una nueva función objetivo derivada de medir la similitud que tiene un círculo candidato con un glóbulo blanco, basándose en la información que tiene la imagen de bordes y la imagen segmentada. Esta nueva función objetivo toma en consideración la información que provee el mapa de bordes y también la relación entre los píxeles que se encuentran dentro del círculo candidato el cual está presente en la imagen segmentada, con esto se valida la existencia de los glóbulos blancos. Esta nueva ecuación  $J(C)$  se define de la siguiente manera:

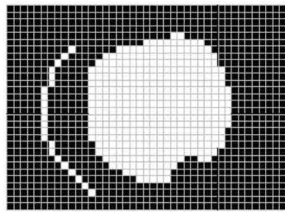
$$J_{Nueva}(C) = 2 - \frac{\sum_{v=1}^{Ns} E(h_v)}{Ns} - \frac{Wp}{Bp} \quad (3.28)$$

donde  $h_v$  y  $Ns$  mantienen el mismo significado que en la ecuación (3.26) y  $Wp$  es la cantidad de píxeles blancos que se encuentran dentro del círculo candidato representado por  $C$ . Del mismo modo  $Bp$  corresponde al número total de píxeles en negro que se encuentran dentro de la circunferencia definida por  $C$ . La función objetivo de la ecuación (3.28) es

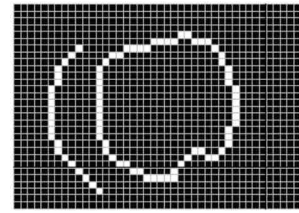
aplicada al mapa de bordes de la imagen, es por esto que sólo se consideran píxeles en blanco para calcular el valor de  $J_{Nueva}(\mathbf{C})$ . Para ilustrar la funcionalidad de la nueva función objetivo, la Figura 3.21 presenta el procedimiento de detección, el cual considera una imagen compleja, esto debido a que en el proceso de adquisición de la muestra sanguínea se generó una deformación del glóbulo blanco. La Figura 3.21 (a) muestra la imagen original de frotis sanguíneo que contiene un glóbulo blanco y una mancha producida por el proceso de coloración de las células. Las Figuras 3.21 (b) y (c) representan la imagen segmentada y el mapa de bordes respectivamente. Ya que la mancha contenida en la imagen del original posee propiedades similares a las de un glóbulo blanco, ésta se mantiene como parte de la imagen segmentada y del mapa de bordes. Siendo ésta una inconsistencia, si se usa el detector de círculos basado en EMO sin ninguna modificación, se producen grandes errores de detección. La Figura 3.21 (d) presenta el resultado de la detección que se obtiene sin modificar la función objetivo, que considera sólo el número de coincidencias entre el círculo candidato y el mapa de bordes. En este mismo ejemplo el círculo detectado tiene una coincidencia de 37 píxeles diferentes en el mapa de bordes, tal coincidencia es considerada la mejor posible bajo las restricciones de la función objetivo original. Por otro lado, cuando se usa la función objetivo modificada en el proceso de reconocimiento de células sanguíneas, la precisión y robustez mejoran significativamente. Mediante el uso de esta nueva función objetivo, la información proveniente de la imagen segmentada es empleada para refinar la solución que provee el mapa de bordes en base a las coincidencias. La Figura 3.21 (d) presenta la detección resultante producida por el algoritmo basado en EMO usando la ecuación (3.28). En esta figura, el círculo detectado se ajusta al mapa de bordes con sólo 32 píxeles. Sin embargo, se considera la mejor opción debido a la relación de los píxeles que componen el área del círculo. Finalmente, la Figura 3.21 (f) muestra la detección final sobre la imagen de frotis sanguíneo original.



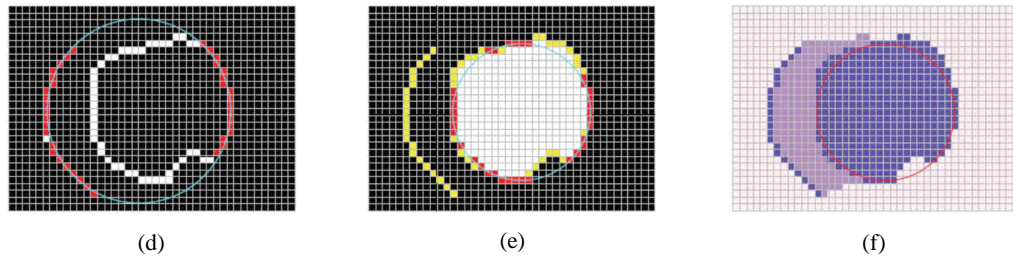
(a)



(b)



(c)



**Figura 3.21:** Proceso de detección de glóbulos blancos. (a) imagen de frotis sanguíneo, (b) imagen segmentada, (c) mapa de bordes, (d) círculo detectado usando la función objetivo original, los puntos rojos muestran la coincidencia entre el círculo candidato y el mapa de bordes. (e) círculo detectado usando la nueva función objetivo, los puntos amarillos representan los bordes sin coincidencia. (f) resultado final.

La Tabla 3.15 presenta los parámetros para el algoritmo EMO usados para la detección de células sanguíneas. Dichos parámetros son los mismos para todas las imágenes de prueba y los experimentos definidos.

$m$	$n$	$MAXITER$	$\delta$	$LSITER$
50	3	5	4	4

**Tabla 3.15:** Parámetros del algoritmo EMO usados para la detección de leucocitos en imágenes médicas.

Bajo tales suposiciones el proceso completo para detectar glóbulos blancos se sintetiza de la siguiente manera:

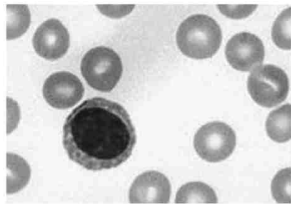
- Paso 1:** Segmentar los leucocitos usando el algoritmo DEM.
- Paso 2:** Obtener el mapa de bordes de la imagen segmentada usando el detector de bordes morfológico.
- Paso 3:** Aplicar el detector de círculos basado en EMO (sección 3.6) sobre el mapa de bordes usando la nueva función objetivo, se deben almacenar los mejores círculos encontrados.
- Paso 4:** Definir los valores de los parámetros para cada círculo que identifican los glóbulos blancos.

### 3.7.3 Un ejemplo numérico

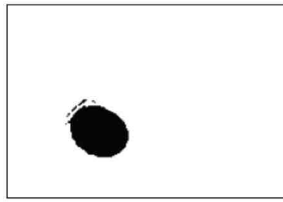
Con el propósito de ilustrarla operación del algoritmo paso a paso, se propone un ejemplo numérico para la detección de un único leucocito en una imagen simple. La Figura 3.22 (a) muestra la imagen usada como ejemplo. Después de aplicar la operación de umbral, el glóbulo blanco es localizado, además de algunos otros píxeles que son solamente ruido, Figura 3.22 (b). Entonces, el mapa de bordes es calculado y almacenado píxel por píxel

dentro del vector  $\mathbf{P}$ . La Figura 3.22 (c) muestra la imagen resultante después de aplicar tal procedimiento.

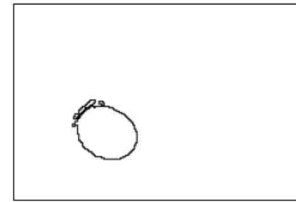
El detector de círculos basado en EMO se ejecuta usando la información del mapa de bordes y la imagen segmentada. Con fines ilustrativos se considera una población de sólo tres partículas. Como todos los enfoques evolutivos, EMO es un optimizador basado en población que al principio ataca el problema muestreando el espacio de búsqueda en múltiples partículas aleatorias. Al tomar tres píxeles aleatorios del vector  $\mathbf{P}$ , se construyen tres diferentes partículas. La Figura 3.22 (d) describe la distribución inicial de dichas partículas. La partícula  $\mathbf{C}_{2,0}$  mantiene el mejor valor de afinidad  $J_{nueva}(\mathbf{C}_{2,0})$ , lo que significa que posee una mejor coincidencia en el mapa de bordes y además posee una buena relación entre los píxeles que se encuentran dentro de la circunferencia, razón por la cual se considera la mejor partícula  $\mathbf{C}^B$ . Entonces, se calcula la carga de cada partícula usando la ecuación (3.2) y las fuerzas ejercidas también son calculadas de acuerdo a las ecuaciones (3.3 y 3.4). La Figura 3.22 (e) muestra las fuerzas ejercidas sobre la partícula  $\mathbf{C}_{3,0}$ , ya que esta partícula es considerada la peor, en términos de la función objetivo, es atraída por las partículas  $\mathbf{C}_{1,0}$  y  $\mathbf{C}_{2,0}$ . Mientras que  $F_{3,1}$  y  $F_{3,2}$  representan las fuerzas de atracción existentes entre  $\mathbf{C}_{3,0}$  con respecto a  $\mathbf{C}_{1,0}$  y  $\mathbf{C}_{2,0}$  donde  $F_3$  es la fuerza resultante. La posición de  $\mathbf{C}_{3,0}$  se modifica usando la fuerza ejercida sobre ella  $F_3$ , de acuerdo al Algoritmo 3.4. La Figura 3.22 (f) describe la nueva posición  $\mathbf{C}_{3,1}$  de la partícula inicial  $\mathbf{C}_{3,0}$  (el segundo subíndice hace referencia al número de iteración). Si se aplica el mismo procedimiento sobre todas las partículas excepto  $\mathbf{C}_{2,0}$  se obtienen las posiciones que se presentan en la Figura 3.22 (g). Por lo tanto, después de 20 iteraciones todas las partículas convergen a la misma posición presentada en la Figura 3.22 (h) y finalmente la célula sanguínea identificada por el mejor círculo se muestra en la Figura 3.22 (i).



(a)

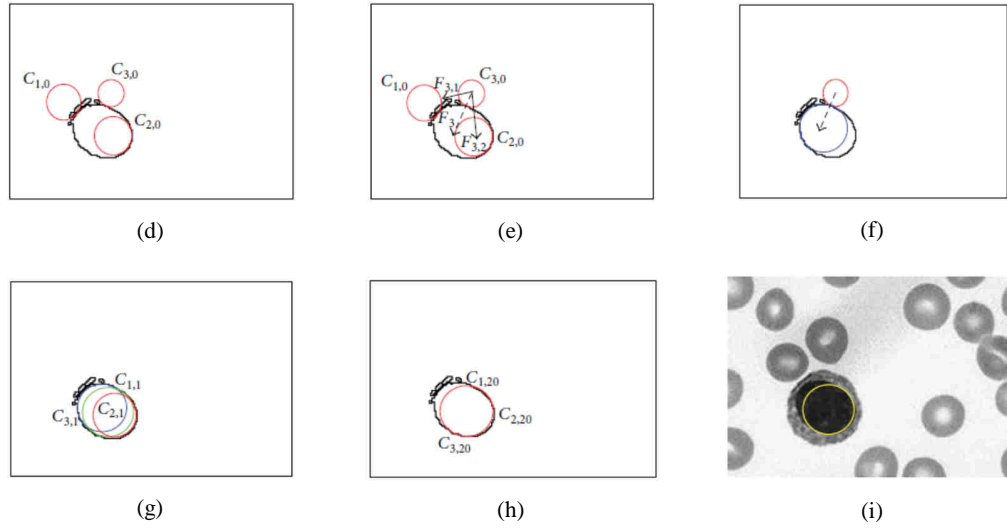


(b)



(c)

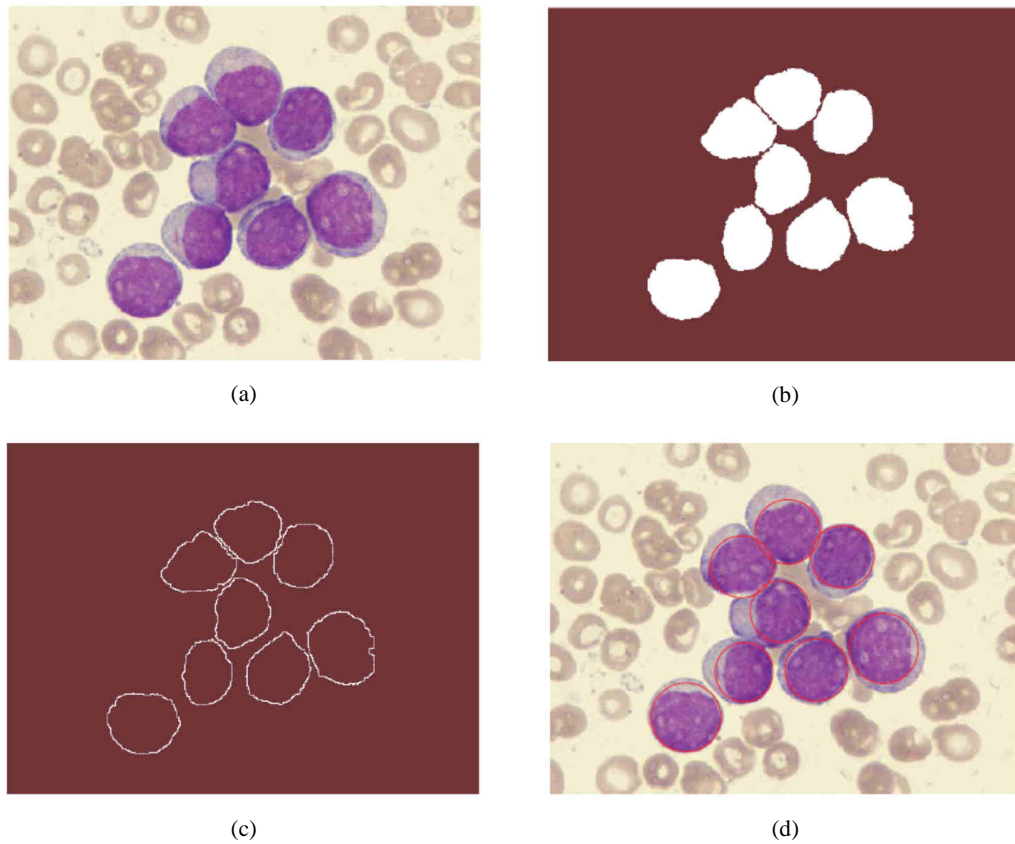




**Figura 3.22:** Ejemplo numérico de detección de una única célula sanguínea: (a) imagen usada como ejemplo, (b) imagen segmentada, (c) mapa de bordes, (d) partículas iniciales, (e) fuerzas ejercidas sobre  $C_{3,0}$ , (f) nueva posición de  $C_{3,0}$ , (g) posición de las partículas después de la primer generación, (h) configuración final de las partículas después de 20 generaciones, (i) resultado final sobre la imagen original.

#### 3.7.4 Resultados experimentales de la detección de glóbulos blancos usando EMO

Las pruebas experimentales se han desarrollado con el objetivo de evaluar el rendimiento del detector propuesto de glóbulos blancos. Dicho procedimiento fue probado sobre imágenes de microscopio procedentes de muestras de frotis sanguíneo de  $600 \times 500$  píxeles de resolución, éstas sirven como apoyo para el diagnóstico de leucemia. Las imágenes muestran diversas condiciones complejas entre las que destacan células deformadas y solapamiento con oclusiones parciales. La robustez del algoritmo ha sido probando bajo tales condiciones. La Figura 3.23 (a) muestra un ejemplo de las imágenes usadas en la prueba. Concretamente ésta, fue usada como imagen de entrada para el detector de glóbulos blancos. La Figura 3.23 (b) presenta las células sanguíneas segmentadas y obtenidas a partir de la aplicación del algoritmo DEM. Las Figuras 3.23 (c) y (d) presentan el mapa de bordes y los leucocitos detectados, respectivamente. Los resultados demuestran que el método propuesto puede detectar efectivamente las células sanguíneas a pesar de la oclusión, deformación o solapamiento, que en ocasiones presentan las células. Otros parámetros pueden ser también calculados por el algoritmo: el área total cubierta por los glóbulos blancos y la relación entre diferentes tamaños de células. En la Figura 3.24 se muestra otro ejemplo del proceso completo de identificación de leucocitos para otra imagen del conjunto de pruebas utilizado.



**Figura 3.23:** Imágenes resultantes de la primera prueba aplicando el detector de glóbulos sanguíneos basado en EMO: (a) imagen original, (b) imagen segmentada por el algoritmo DEM, (c) mapa de bordes y (d) leucocitos detectados.

#### 3.7.4.1 Comparaciones con otros métodos

Un conjunto completo de imágenes de frotis sanguíneo es usado para probar el rendimiento del enfoque propuesto. El detector propuesto basado en EMO se ha aplicado para probar su rendimiento en las imágenes propuestas frente a otros algoritmos empleados con la misma finalidad, como son: los enfoques de vectores soporte de contorno (*boundary support vectors*, BSVs) [73], el método de Otsu iterativo (IO) [74], el algoritmo de Wang [75] y el detector basado en algoritmos genéticos (BGA) [76]. En todos los casos, los se sintonizan de acuerdo al conjunto de valores con los que originalmente han sido propuestos en sus propias referencias.

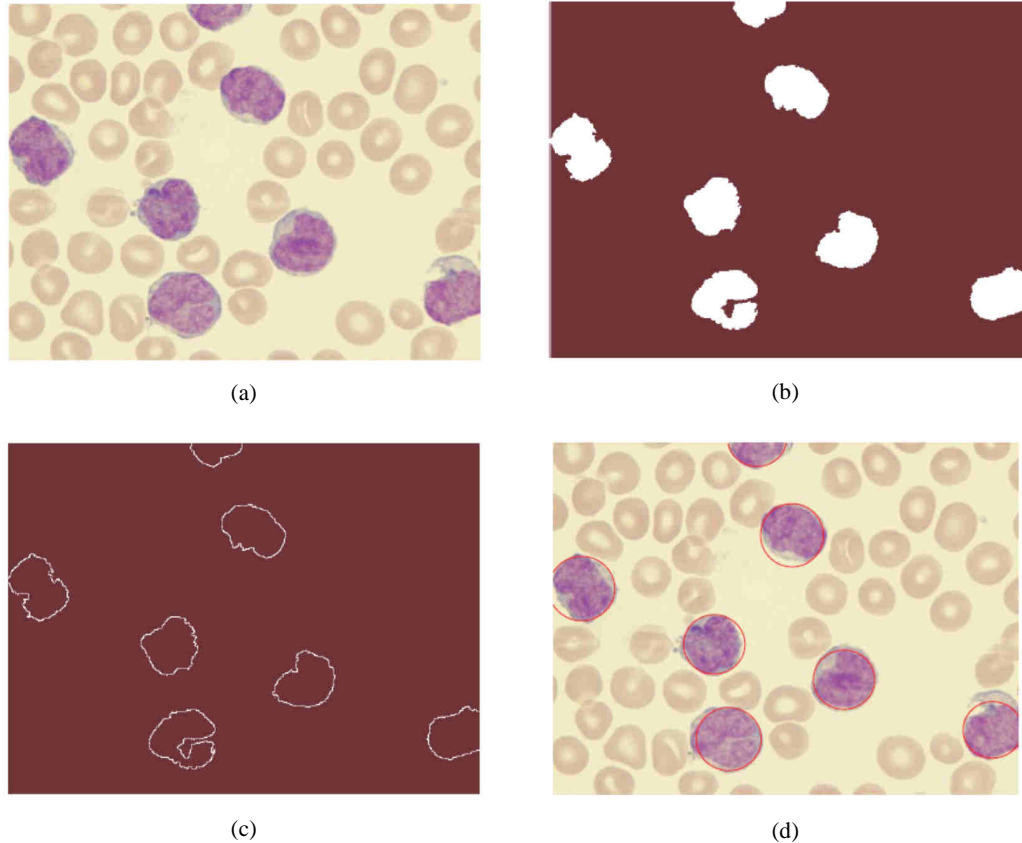
*Comparación en la detección:* para evaluar el rendimiento del método propuesto para la detección de células sanguíneas, en la Tabla 3.16 se muestra un tabulador comparativo del rendimiento en la detección de leucocitos, para los métodos BSV, IO,

Wang, BGA y el enfoque basado en EMO, todo esto en términos de rangos de detección y falsas alarmas. El conjunto de datos experimentales incluye 30 imágenes, las cuales han sido recolectadas de la librería de referencia Cella vision [181]. Tales imágenes contienen 426 leucocitos, divididos de acuerdo a las condiciones del frotis en: 222 leucocitos brillantes y 204 leucocitos oscuros, estos valores han sido obtenidos por el conteo realizado por expertos humanos y sirven como base para todos los experimentos. Para la comparación del índice de detección (DR) se define como la relación entre el número de leucocitos detectados correctamente y el número de leucocitos determinado por el experto. El índice de falsa alarma (FAR) se define como la relación entre la cantidad de objetos que no son leucocitos, que son detectados erróneamente y el número de leucocitos determinados por el experto.

Tipo de leucocito	Método	Leucocitos detectados	Leucocitos perdidos	Falsas alarmas	DR (%)	FAR (%)
<b>Leucocitos brillantes (222)</b>	BSV	104	118	67	46.85	30.18
	IO	175	47	55	78.83	24.77
	Wang	186	36	42	83.78	18.92
	BGA	177	45	22	79.73	9.91
	EMO	211	11	10	95.04	4.50
<b>Leucocitos oscuros (204)</b>	BSV	98	106	54	48.04	26.47
	IO	166	38	49	81.37	24.02
	Wang	181	23	38	88.72	18.63
	BGA	170	134	19	83.33	9.31
	EMO	200	4	6	98.04	2.94
<b>Total</b>	BSV	202	224	121	47.42	28.40
	IO	341	85	104	80.05	24.41
	Wang	367	59	80	86.15	18.78
	BGA	347	79	41	81.45	9.62
	EMO	411	15	16	96.48	3.75

**Tabla 3.16:** Comparación del rendimiento en la detección de leucocitos para los enfoques BSV, IO, Wang, BGA y el método propuesto basado en EMO.

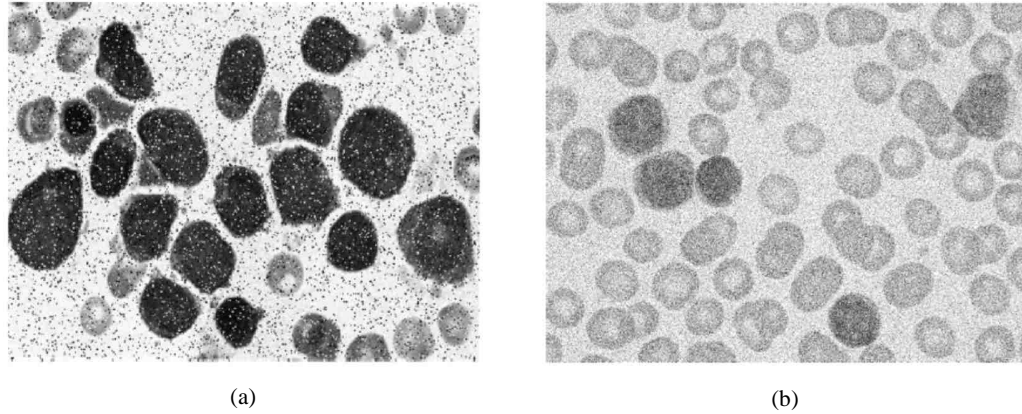
Los resultados experimentales muestran que el método EMO propuesto, tras su comparación frente a otros algoritmos de detección de glóbulos sanguíneos, alcanza resultados favorables como son el 96.48% de exactitud en la detección de leucocitos con un 3.75% de rango de falsas alarmas.



**Figura 3.24:** Imágenes resultantes de la primera prueba aplicando el detector de glóbulos sanguíneos basado en EMO: (a) imagen original, (b) imagen segmentada por el algoritmo DEM, (c) mapa de bordes y (d) leucocitos detectados.

*Comparaciones de robustez:* las imágenes de frotis sanguíneo son comúnmente deterioradas por ruido debido a varias fuentes de interferencia y otros fenómenos que afectan los procesos de medición en las imágenes y en los sistemas de adquisición de datos. Por lo tanto, los resultados de la detección dependen de la habilidad de los algoritmos para hacer frente a diferentes tipos de ruido. Para demostrar la robustez en la detección de glóbulos blancos, el enfoque propuesto basado en EMO es comparado con BSV, IO, Wang y BGA en condiciones de altos niveles de ruido. En esta prueba se estudian dos experimentos diferentes, en la primera investigación se explora el rendimiento de cada algoritmo cuando la tarea de detección se lleva a cabo sobre las imágenes dañadas por ruido

sal y pimienta. El segundo experimento considera imágenes contaminadas por ruido Gaussiano. Ambos tipos de ruido han sido seleccionados, porque son los más compatibles con el ruido que puede encontrarse comúnmente en las imágenes de frotis sanguíneo [182].



**Figura 3.25:** Ejemplos de imágenes incluidas en el conjunto de prueba para la comparación de robustez. (a) Imagen contaminada con el 10% de ruido sal y pimienta y (b) imagen con  $\sigma=10$  de ruido Gaussiano.

La comparación considera el conjunto de 30 imágenes empleado para el experimento anterior, las cuales contienen 426 leucocitos. El ruido que se ha añadido se genera mediante la función correspondiente del *Image Processing toolbox* de Matlab, considerando dos niveles de ruido 5% y 10% para el ruido sal y pimienta, mientras que para el ruido Gaussiano se usan los valores:  $\sigma = 5$  y  $\sigma = 10$ . La Figura 3.25 muestra un ejemplo de sólo dos imágenes con diferentes tipos de ruido. Los resultados en términos del índice de detección (DR) y el índice de falsa alarma (FAR) son reportados para cada tipo de ruido en las Tablas 3.17 y 3.18. Dichos resultados muestran que el algoritmo EMO propuesto presenta un mejor rendimiento en la detección de imágenes contaminadas, alcanzando en el peor caso un DR de 87.79% para ruido sal y pimienta y un DR de 89.20% para ruido Gaussiano. Por otro lado, el detector basado en EMO posee el mejor rendimiento de degradación, presentando un FAR de 8.21% y 7.51% para cada tipo de ruido.

*Comparaciones de estabilidad:* con la finalidad de comparar el rendimiento en la estabilidad del método propuesto, los resultados se comparan con los reportados por Wang et al. [75], cuyo trabajo se considera como una de las técnicas más precisas para la detección de glóbulos blancos.

Nivel de ruido	Método	Leucocitos detectados	Leucocitos perdidos	Falsas alarmas	DR (%)	FAR (%)
5% de ruido sal y pimienta, 426 leucocitos	BSV	148	278	114	34.74	26.76
	IO	270	156	106	63.38	24.88
	Wang	250	176	118	58.68	27.70
	BGA	306	120	103	71.83	24.18
	EMO	390	36	30	91.55	7.04
10% de ruido sal y pimienta, 426 leucocitos	BSV	101	325	120	23.71	28.17
	IO	240	186	78	56.34	18.31
	Wang	184	242	123	43.19	28.87
	BGA	294	132	83	69.01	19.48
	EMO	374	52	35	87.79	8.21

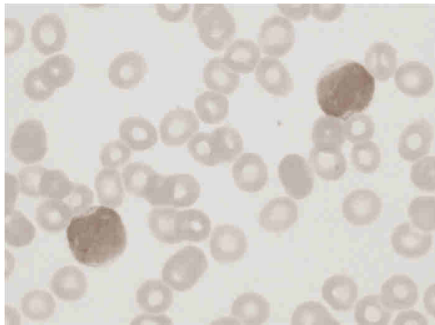
**Tabla 3.17:** Comparación en la detección de leucocitos entre los enfoques BSV, IO, Wang, BGA y el método propuesto basado en EMO, considerando un conjunto de 30 imágenes contaminadas con diferentes niveles de ruido sal y pimienta.

Nivel de ruido	Método	Leucocitos detectados	Leucocitos perdidos	Falsas alarmas	DR (%)	FAR (%)
$\sigma=5$ de ruido Gaussiano, 426 leucocitos	BSV	172	254	77	40.37	18.07
	IO	309	117	71	72.53	16.67
	Wang	301	125	65	70.66	15.26
	BGA	345	81	61	80.98	14.32
	EMO	397	29	21	93.19	4.93
$\sigma=10$ de ruido Gaussiano, 426 leucocitos	BSV	143	283	106	33.57	24.88
	IO	281	145	89	65.96	20.89
	Wang	264	162	102	61.97	23.94
	BGA	308	118	85	72.30	19.95
	EMO	380	46	32	89.20	7.51

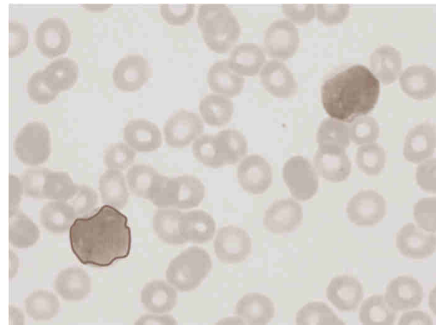
**Tabla 3.18:** Comparación en la detección de leucocitos entre los enfoques BSV, IO, Wang, BGA y el método propuesto basado en EMO, considerando un conjunto de 30 imágenes contaminadas con diferentes niveles de Gaussianos.

El algoritmo de Wang es un método de minimización de energía, el cual se guía por elementos de restricción internos e influenciados por fuerzas externas de la imagen, produciendo la segmentación de glóbulos blancos según un contorno cerrado. El enfoque de Wang usa como fuerzas externas un tipo de información específica, la cual es usualmente representada por el gradiente de la magnitud de la imagen. Por lo tanto, el contorno es atraído por los píxeles con valores de gradiente grandes, que son bordes considerados fuertes. En cada iteración, el método de Wang encuentra una nueva configuración de contorno, que minimiza la energía correspondiente a las fuerzas externas y a los elementos que imponen las restricciones.

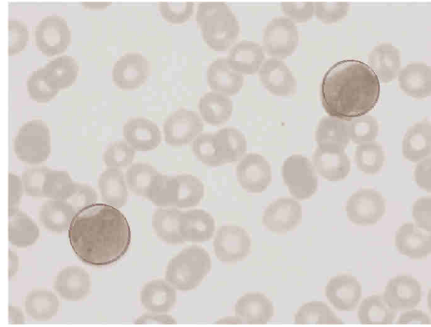
En la comparación, la estructura de la red neuronal celular y sus parámetros operacionales, se corresponden con el algoritmo de Wang, siguiendo la configuración sugerida por sus autores, mientras que los parámetros para el algoritmo EMO son tomados de la Tabla 3.15. La Figura 3.26 muestra el rendimiento de ambos métodos considerando una imagen de prueba con sólo dos leucocitos. El método de Wang usa información de gradiente para encontrar apropiadamente una nueva configuración de contorno, además necesita ser ejecutado iterativamente para detectar cada estructura. La Figura 3.26 (b) muestra la evidencia de los resultados toda vez que el enfoque de Wang ha sido aplicado considerando sólo 200 iteraciones. Por otro lado, la Figura 3.26 (c) presenta los resultados después de aplicar el método EMO propuesto para detectar los leucocitos.



(a)



(b)



(c)

**Figura 3.26:** Comparación de los métodos de Wang y EMO para detección de células sanguíneas en imágenes medicas. (a) Imagen original, (b) resultados de la detección usando en el algoritmo de Wang, (c) detección tras aplicar el algoritmo EMO.

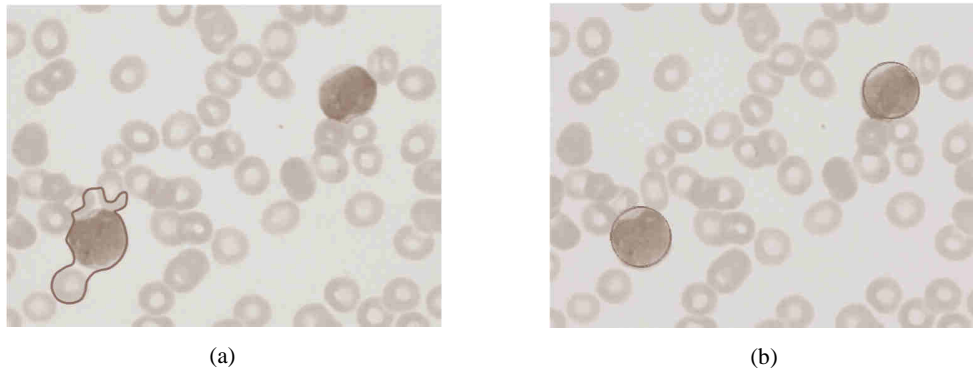
El algoritmo de Wang usa las redes neuronales celulares (*fuzzy celular neural network*, FCNN) como optimizador. Este enfoque emplea la información de gradiente y los estados internos para encontrar una mejor configuración de contornos. En cada iteración, las FCNN tratan también como puntos de contorno las nuevas y diferentes posiciones de píxeles, que deben ser localizadas próximas al contorno original. Tal hecho puede ser causa de que el contorno solución se mantenga atrapado en un mínimo local. Para evitar este problema el método de Wang aplica un considerable número de iteraciones de manera tal que sea factible encontrar una configuración de contorno cercana. Sin embargo, cuando el número de iteraciones se incrementa, la posibilidad de cubrir otras estructuras también aumenta. Entonces, si la imagen tiene un fondo complejo, como ocurre con las imágenes de frotis, el método se confunde y no resulta fácil encontrar la mejor configuración de contorno. Por consiguiente, un inconveniente del método de Wang se relaciona con el óptimo número de iteraciones. Tal valor debe ser determinado experimentalmente ya que depende del contexto de la imagen y de su complejidad. La Figura 3.27 (a) presenta el resultado de aplicar 400 ciclos del algoritmo de Wang, mientras que la Figura 3.27 (b) muestra la detección de las mismas células después de 1000 iteraciones usando el algoritmo EMO propuesto. A partir de los resultados mostrados en la Figura 3.27 (a) se puede apreciar que el contorno producido por el algoritmo de Wang se degenera conforme al proceso iterativo progresa, cubriendo erróneamente otras formas que se encuentran cerca de la célula que se pretende detectar. Para realizar una comparación de la precisión de ambos enfoques, el área del glóbulo blanco estimada, que ha sido aproximada por ambos algoritmos, se compara con el tamaño original de la misma célula, considerando diferentes grados de evolución, esto se refiere al número de ciclos de cada algoritmo. La comparación se realiza sobre un único leucocito, debido a que el método de Wang sólo es capaz de



detectar una única forma. La Tabla 3.19 presenta el promedio de los resultados tras realizar 20 iteraciones para cada experimento.

Algoritmo	Iteraciones	Error (%)
Wang	60	70.0
	200	1.0
	400	121.0
EMO propuesto	60	8.22
	200	10.1
	400	10.8

**Tabla 3.19:** Error en la estimación del tamaño de las células después aplicar el algoritmo EMO y el método de Wang para detectar un leucocito contenido dentro de una imagen de frotis sanguíneo. El error es promediado a partir de 20 experimentos.



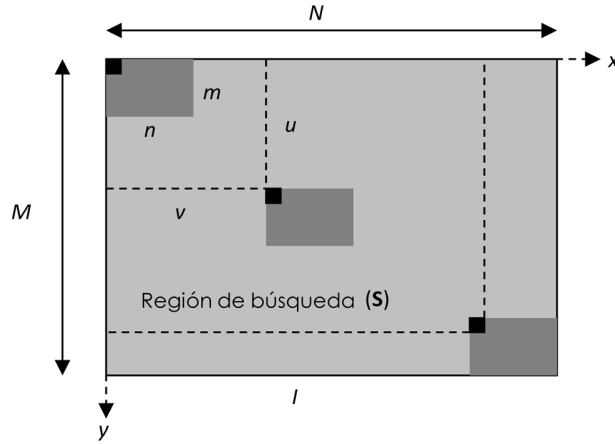
**Figura 3.27:** Resultados de comparación para la detección de glóbulos blancos mostrando: (a) el algoritmo de Wang después de 400 ciclos y (b) el detector basado en EMO considerando 1000 ciclos.

### 3.8 Localización y detección de objetos por medio de plantillas

Para iniciar el proceso de localización de objetos a partir de una imagen de referencia (patrón o plantilla)  $R$  sobre una imagen considerablemente más grande  $I$ , la tarea consiste en encontrar las posiciones en la imagen  $I$  cuyas regiones coincidan o al menos se asimilen con  $R$ .

Definiendo la plantilla desplazada como  $R_{u,v}(x, y) = R(x - u, y - v)$  donde  $x$  e  $y$  son las coordenadas de posición de la plantilla  $R$ , la cual se desplaza por medio de la distancia  $(u, v)$  en las direcciones horizontal y vertical, respectivamente. Entonces, el

problema de localización de objetos (que se ilustra en la Figura 3.28) se puede resumir como sigue: considerando la imagen fuente  $I$  y la imagen de referencia  $R$ , encontrar el desplazamiento  $(u, v)$  con el cual la región de búsqueda  $S$  cuente con la máxima similitud entre la imagen de referencia desplazada  $R_{u,v}(x, y)$  y la correspondiente sub-imagen de  $I$ .



**Figura 3.28:** Geometría del problema de localización y reconocimiento de objetos mediante plantillas.

Para localizar objetos exitosamente mediante el uso de plantillas, además de seguir la metodología previamente planteada, es necesario definir dos problemas más. El primero consiste en determinar un valor de similitud que ayudará a validar cuándo ha ocurrido una coincidencia entre el objeto de la imagen de referencia y de la imagen fuente. El segundo, consiste en desarrollar una estrategia de búsqueda eficiente para encontrar el desplazamiento óptimo. A pesar de que existen numerosas métricas para evaluar la similitud entre dos imágenes, de acuerdo con la literatura, las más importantes son las tres siguientes: la suma de diferencias absoluta (SAD), la suma de diferencias cuadradas (SSD) y la correlación cruzada normalizada (NCC). El cálculo de estas métricas suele ser costoso en términos computacionales y representa la operación que más tiempo consume en todo el proceso [183]. Aunque el NCC, SAD y SSD permiten una adecuada medición de la similitud entre dos imágenes, el NCC es el más robusto, razón por la cual se emplea ampliamente en diversas aplicaciones [183].

El valor de NCC que existe entre una imagen dada  $I$  de tamaño  $M \times N$  y una imagen plantilla (o patrón)  $R$  de tamaño  $m \times n$ , con un desplazamiento de  $(u, v)$ , se obtiene por medio de la siguiente ecuación:

$$NCC(u, v) = \frac{\sum_{i=1}^m \sum_{j=1}^n [I(u+i, v+j) - \bar{I}(u, v)] \cdot [R(i, j) - \bar{R}]}{\sqrt{\left[ \sum_{i=1}^m \sum_{j=1}^n I(u+i, v+j) - \bar{I}(u, v) \right]^2 \cdot \left[ \sum_{i=1}^m \sum_{j=1}^n R(i, j) - \bar{R} \right]^2}} \quad (3.29)$$

donde  $\bar{I}(u, v)$  es la intensidad promedio en escala de grises de la imagen fuente, para la región que coincide con la imagen plantilla  $R$ , donde  $\bar{R}$  es la intensidad promedio en escala de grises para la plantilla. Estos valores se definen como:

$$\bar{I}(u, v) = \frac{1}{m \cdot n} \sum_{i=1}^m \sum_{j=1}^n I(u+i, v+j), \quad \bar{R} = \frac{1}{m \cdot n} \sum_{i=1}^m \sum_{j=1}^m R(i, j) \quad (3.30)$$

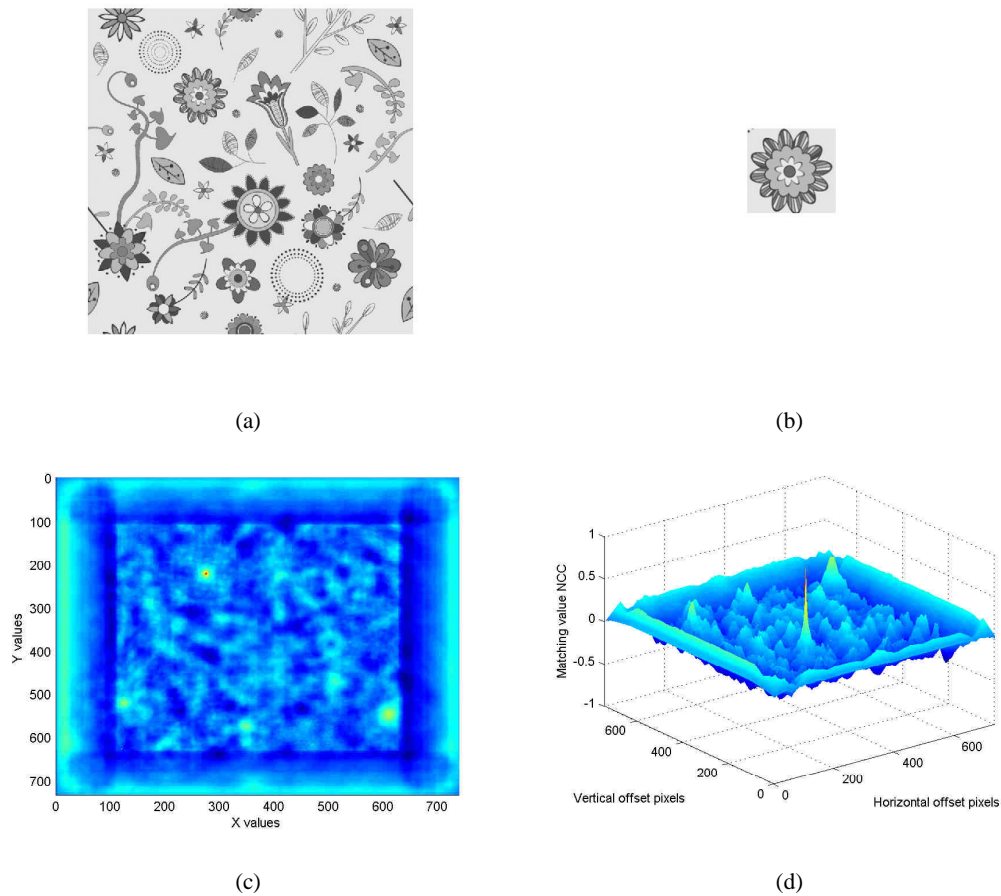
La operación de NCC da como resultado valores dentro del intervalo  $[-1, 1]$ . Entonces, si  $NCC=1$  se tendrá la mejor similitud entre las imágenes, por el contrario un valor de  $NCC=-1$  demuestra que ambas imágenes son completamente diferentes. Un caso especial puede encontrarse cuando una plantilla es comparada con una región con menos iluminación en la imagen fuente, es decir que dicha región contiene píxeles cuyos valores son iguales o cercanos a cero. En estos casos, para evitar la indeterminación en la ecuación (3.29), el valor de NCC se configura a un valor predeterminado de -1. De acuerdo con el valor de NCC, el punto  $(u, v)$  que presenta la mayor similitud posible entre  $R$  e  $I$  se define como:

$$(u, v) = \arg \max_{(\hat{u}, \hat{v}) \in S} NCC(\hat{u}, \hat{v}) \quad (3.31)$$

donde  $S = \{(\hat{u}, \hat{v}) \mid 1 \leq \hat{u} \leq M - m, 1 \leq \hat{v} \leq N - n\}$ . Se requiere de una eficiente estrategia de búsqueda para determinar la posición  $(u, v)$  donde se encuentra la máxima similitud. El algoritmo más simple para esta tarea, es el de búsqueda exhaustiva ya que proporciona la detección óptima con respecto al valor de NCC revisando todos los píxeles candidatos uno por uno. Desafortunadamente, este tipo de búsqueda en combinación con el cálculo de la NCC para cada punto de la imagen, es extremadamente costoso en términos computacionales, lo cual restringe seriamente su uso en aplicaciones de procesamiento de imagen.

La Figura 3.29 ilustra el proceso de localización y detección de objetos mediante el uso de plantillas, considerando la Figura 3.29 (a) como la imagen fuente y la Figura 3.29 (b) como la imagen plantilla o patrón, la cual es similar pero no idéntica al patrón identificado en la imagen fuente. La Figura 3.29 (c) muestra los valores NCC (en color) calculados para

todas las localidades de la región de búsqueda  $S$  usando la estrategia de búsqueda exhaustiva. Por otro lado, en la Figura 3.29 (d) se presenta la superficie de búsqueda de la función NCC, la cual es altamente multimodal, esta característica es propia del problema. Las Figuras 3.29 (c) y (d) muestran que los valores de la superficie NCC tienen una gran cantidad de valores máximos locales y sólo un máximo global. Por tal motivo, los métodos tradicionales de optimización y en particular los que están basados en gradiente, pueden ser atrapados por estos valores locales.



**Figura 3.29:** Proceso de localización y detección de objetos en imágenes mediante el uso de plantillas, (a) imagen fuente, (b) imagen plantilla, (c) valores de NCC codificados en color y (d) superficie multimodal de NCC.

### 3.9 Modificaciones del algoritmo EMO para dar solución al problema de localización y detección de objetos mediante el uso de plantillas

Como parte de la investigación, se propone el uso de una versión modificada del algoritmo EMO como estrategia de búsqueda para resolver el problema de localización y

detección de objetos usando plantillas. La nueva implementación evita dos inconvenientes tradicionales del algoritmo EMO: el uso de un gran número de evaluaciones de la función objetivo y la reevaluación de individuos que ya han sido analizados. Aquí se describen dos modificaciones para mejorar el rendimiento de EMO. Tales modificaciones involucran la construcción de un nuevo proceso de búsqueda local (BL) y la incorporación de una memoria para almacenar los valores de la función objetivo que ya han sido calculados. Como resultado de estas modificaciones, el nuevo algoritmo EMO puede reducir substancialmente el número de evaluaciones de la función objetivo (para este caso NCC), preservando las buenas capacidades de búsqueda que posee el método original.

### 3.9.1 *El nuevo proceso de búsqueda local para EMO*

La fase de explotación dentro del algoritmo EMO, es representada por la búsqueda local. En términos generales, en la explotación se lleva a cabo el refinamiento de los individuos existentes, esto se realiza en un pequeño vecindario con el objetivo de mejorar su calidad como solución. En la literatura referente [18, 23], se proponen dos enfoques de BL para el algoritmo EMO: la búsqueda local aplicada a todos los elementos de la población y la búsqueda local aplicada sólo al mejor elemento de la población. Para el primer caso, se ha comprobado que EMO presenta las propiedades de convergencia más óptimas hasta alcanzar la optimización global del problema [18]. Sin embargo, bajo estas circunstancias, algunos estudios experimentales [14, 23] muestran que la BL requiere más del 80% del tiempo de cómputo en todo el proceso de optimización en EMO. Por otra parte, los resultados experimentales demuestran que si la BL se aplica sólo a la mejor partícula actual, se debilitan seriamente las capacidades de búsqueda del algoritmo EMO [23, 184], principalmente en problemas de optimización complejos. Bajo tales condiciones, se propone un nuevo método de búsqueda local para reducir el tiempo de cómputo (número de evaluaciones de la función objetivo o NCC), evitando degradar alguna de las capacidades de búsqueda del EMO. A diferencia de otros enfoques propuestos [18, 23, 184], el nuevo método de búsqueda local considera una compensación entre el uso de todas las partículas y sólo la mejor. Este balance se obtiene perturbando únicamente las soluciones más prometedoras de la población.

El nuevo proceso de búsqueda local es una operación selectiva, la cual se aplica sólo a un subconjunto  $E_t$  de la población de partículas modificadas  $Y_t$ , considerando que  $E_t \subseteq Y_t$ . En el nuevo enfoque de BL, primero es necesario ordenar  $Y_t$  de acuerdo a sus

valores de función objetivo y almacenar los elementos ordenados en una población temporal  $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_N\}$ . La idea es considerar sólo las posiciones más prometedoras. En el nuevo mecanismo de BL se crea un sub-espacio  $C_j$  alrededor de cada partícula seleccionada  $\mathbf{b}_j \in \mathbf{B}$  ( $\mathbf{b}_j = \{b_j^1, \dots, b_j^n\}$ ). El tamaño de  $C_j$  depende de un radio de distancia  $e_d$  el cual se define como:

$$e_d = \frac{\sum_{q=1}^n (ub_q - lb_q)}{n} \cdot \beta \quad (3.32)$$

donde  $ub_q$  y  $lb_q$  son los límites inferior y superior en la dimensión  $q$ ,  $n$  es el número de dimensiones del problema de optimización, mientras que  $\beta \in [0, 1]$  es un factor de ajuste. Por lo tanto, los límites de  $C_j$  son modelados como sigue:

$$uss_j^q = b_j^q + e_d, \quad lss_j^q = b_j^q - e_d \quad (3.33)$$

donde  $uss_j^q$  y  $lss_j^q$  son los límites inferior y superior de la  $q$ -ésima dimensión para cada sub-espacio  $C_j$ , respectivamente.

Considerando el sub-espacio  $C_j$  alrededor de cada elemento  $\mathbf{b}_j$  de  $\mathbf{B}$ , se genera aleatoriamente un conjunto de  $h$  nuevas partículas ( $\mathbf{P}_j^h = \{p_j^1, p_j^2, \dots, p_j^h\}$ ), cuyos valores se encuentran dentro de los límites definidos por la ecuación (3.33). Una vez que las  $h$  muestras son generadas, se debe crear la partícula  $\mathbf{x}_{j,t+1}$  de la población  $\mathbf{X}_{t+1}$ . Para calcular  $\mathbf{x}_{j,t+1}$ , se compara la mejor partícula en términos de la función objetivo  $\mathbf{p}_j^{mejor}$  que pertenece a las  $h$  muestras con  $\mathbf{b}_j$ , donde  $\mathbf{p}_j^{mejor} \in [\mathbf{p}_{j,1}, \mathbf{p}_{j,2}, \dots, \mathbf{p}_{j,h}]$ . Si  $\mathbf{p}_j^{mejor}$  es mejor que  $\mathbf{b}_j$  de acuerdo a sus valores de función objetivo,  $\mathbf{x}_{j,t+1}$  se actualiza con  $\mathbf{p}_j^{mejor}$ ; en cualquier otro caso, se selecciona  $\mathbf{b}_j$ . Los elementos de  $\mathbf{B} \subseteq \mathbf{Y}_t$  que no fueron analizados en la nueva búsqueda local, transfieren sus valores correspondientes a la población  $\mathbf{X}_{t+1}$  sin sufrir ninguna modificación.

Como ya se mencionó previamente, la nueva búsqueda local, es un proceso selectivo, el cual sólo se aplica a los mejores elementos de la población. Para ser considerados por la nueva BL, dichos elementos deben cumplir dos condiciones. La

primera es que cada partícula  $\mathbf{b}_j \in \mathbf{B}$  debe ser parte de la primera mitad de la población temporal  $\mathbf{B}$ , la cual se genera después de ordenar  $\mathbf{Y}_t$ . La segunda condición es que dentro de  $C_j$  no debe existir ninguna partícula que tenga un valor mejor de función objetivo que  $\mathbf{b}_j$ . La primera condición asegura que sólo los individuos con los mejores valores en la función objetivo pueden ser considerados para la explotación. Ya que el algoritmo EMO tiende a concentrarse en una determinada solución durante la evolución del proceso de optimización, la segunda condición garantiza que el número de partículas en el proceso de explotación se decrementa en cada iteración. Este comportamiento refleja el hecho de que una gran concentración de partículas en una solución representa un tipo de auto explotación extensiva y bajo estas condiciones no es necesario aplicar la búsqueda local de nuevo.

El enfoque de BL propuesto se usa para explotar las soluciones prominentes. Tal mecanismo de búsqueda local reduce el esfuerzo computacional producido por el método tradicional con el que fue propuesto EMO [13]. El nuevo proceso ordena la población de acuerdo a los valores de la función objetivo y sólo se usan los mejores elementos de la población; sin embargo, dichos elementos deben estar lo suficientemente separados entre sí. La distancia  $e_d$  tiene entonces un doble propósito. Primero, se usa para determinar cuáles partículas son seleccionadas de la población ordenada. Los elementos seleccionados deben estar separados por al menos dos veces el valor de la distancia de explotación  $e_d$ . El segundo propósito de  $e_d$  es crear los sub-espacios de los cuales se tomarán las nuevas partículas. Por lo tanto, dentro de cada sub-espacio, se seleccionan  $h$  muestras aleatorias. De todas las muestras, se selecciona la mejor de acuerdo a su valor en la función objetivo. Si la muestra tiene el mejor valor en comparación con la partícula original, su posición se actualiza; en caso contrario no sufre ninguna modificación.

Ya que el número de muestras seleccionadas es muy pequeño ( $h < 4$ ) en la aplicación que se hace para la búsqueda, el uso de la nueva BL reduce substancialmente el número de evaluaciones de la función objetivo. Por otro lado, el uso de la regla de distancia para descartar las partículas durante el proceso de evolución, garantiza que se creen menos sub-espacios y consecuentemente se evalúen menos muestras. Al final del proceso de optimización, sólo la mejor partícula es explotada, ya que por su cercanía las otras partículas son discriminadas. El nuevo enfoque de búsqueda local se describe por completo en el Algoritmo 3.5.

**Algoritmo 3.5: Nueva Búsqueda Local**


---

```

1:  $\mathbf{B} \leftarrow \text{Ordenar}(\mathbf{Y}_t)$ 
2: for  $p=1$  to  $N/2$  do
3:    $b_j \leftarrow \text{Obtener\_de } \mathbf{B}(b_p)$ 
4:    $C_j \leftarrow \text{Definir\_subespacios\_para } (b_j)$ 
5:    $b_c \leftarrow \text{Obtener\_el\_elemento\_cercano\_a } (b_j)$ 
6:   if  $b_c \in C_j$  then
7:      $\text{bandera1} = 1$ 
8:   end if
9:   if  $f(b_j) \geq f(b_c)$  then
10:     $\text{bandera2} = 1$ 
11:   end if
12:   if  $(\text{no}(\text{bandera1}))$  or  $(\text{bandera1 y bandera2})$  then
13:      $\mathbf{P}_j^m \leftarrow \text{Genera\_m\_muestras } (C_j)$ 
14:      $p_j^{\text{best}} \leftarrow \text{Obtener\_mejor } (\mathbf{P}_j^m)$ 
15:     if  $f(b_j) < f(p_j^{\text{best}})$  then
16:        $x_{j,t+1} = p_j^{\text{best}}$ 
17:     else
18:        $x_{j,t+1} = b_j$ 
19:     end if
20:   end if
21: end for
22: for  $p = N/2 + 1$  to  $N$  do
23:    $b_i \leftarrow \text{Obtener\_de } \mathbf{B}(b_p)$ 
24:    $x_{i,t+1} = b_i$ 
24: end for

```

---

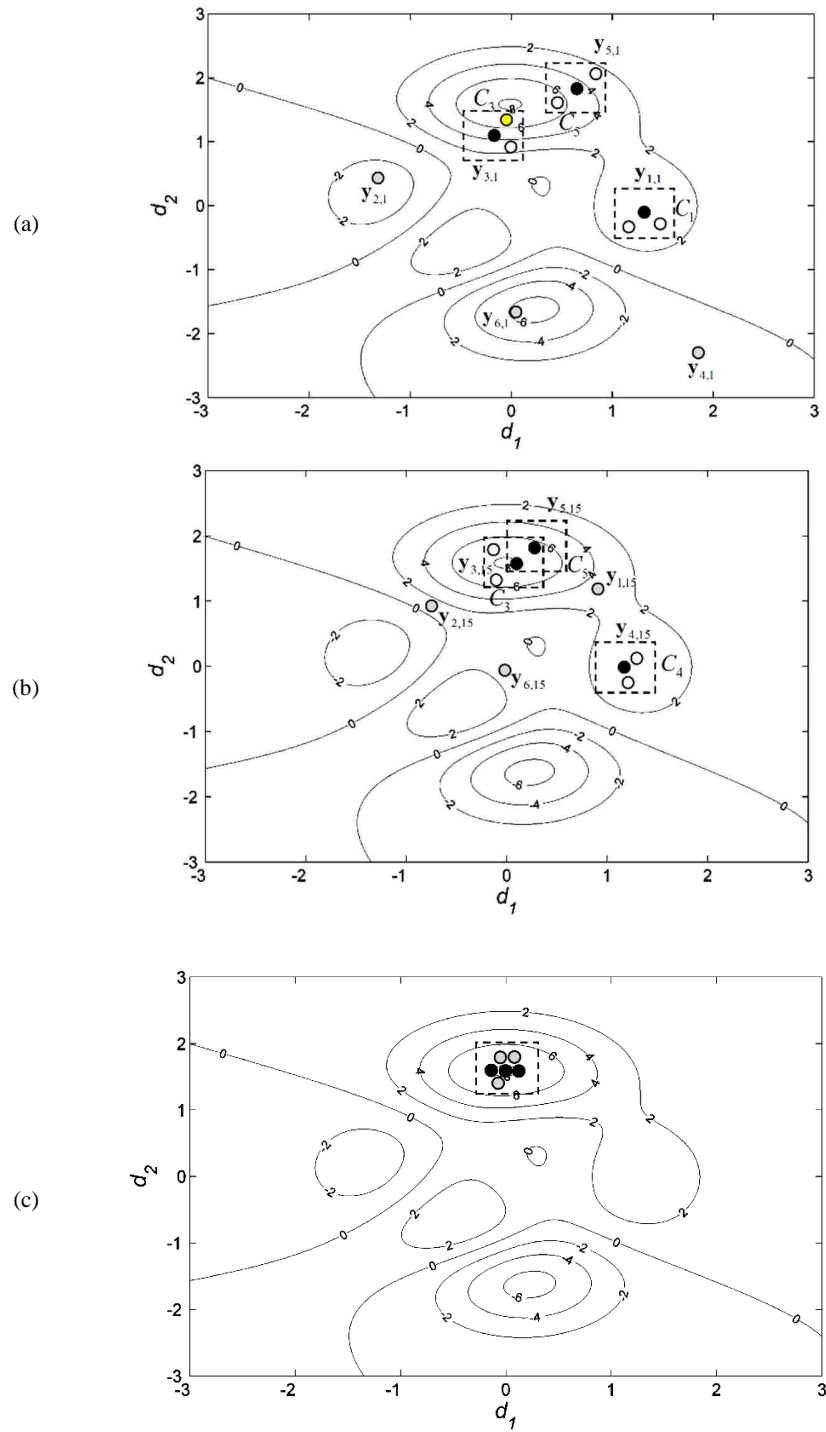
Con fines ilustrativos a continuación se propone un ejemplo numérico que demuestra el funcionamiento del nuevo proceso de búsqueda local, el cual es aplicado a una función sencilla de dos dimensiones. Dicha función considera el intervalo de  $-3 \leq d_1, d_2 \leq 3$  mientras que la función tiene un máximo valor global de 8.1 en la posición  $(0, 1.6)$ . Cabe mencionar, que  $d_1$  y  $d_2$  corresponden a los ejes coordenados (comúnmente  $x$  e  $y$ ). Para este ejemplo, se asume una población modificada  $\mathbf{Y}_t$  de 6 miembros bi-dimensionales ( $N = 6$ ).

En la Figura 3.30 (a) se muestra la configuración inicial del ejemplo propuesto; los puntos negros representan la mitad de las partículas con los mejores valores en la función objetivo (la primera mitad de  $\mathbf{B}$ ) mientras que los puntos grises corresponden al resto de las partículas. De la Figura 3.30 (a) se puede observar que el nuevo proceso de BL se aplica a todas las partículas negras ( $\mathbf{y}_{1,1} = \mathbf{b}_1$ ,  $\mathbf{y}_{3,1} = \mathbf{b}_2$ , e  $\mathbf{y}_{5,1} = \mathbf{b}_3$ ), manteniendo dos nuevas partículas aleatorias (caracterizadas por los puntos blancos  $\mathbf{P}_1^2$ ,  $\mathbf{P}_3^2$  y  $\mathbf{P}_5^2$ ) para cada punto



negro dentro de sus correspondientes sub-espacios ( $C_1$ ,  $C_3$ , y  $C_5$ ). De esta forma, se ejecuta la operación correspondiente sobre  $\mathbf{y}_{1,1}$ ,  $\mathbf{y}_{3,1}$  e  $\mathbf{y}_{5,1}$ , debido a que cumplen con las dos condiciones necesarias. Considerando la partícula  $\mathbf{y}_{3,1}$  en la Figura 3.30 (a). La partícula amarilla corresponde a la mejor ( $\mathbf{p}_3^{mejor}$ ) de las dos partículas generadas aleatoriamente dentro de  $C_3$  y de acuerdo a su valor en la función objetivo. De este modo la partícula  $\mathbf{p}_3^{mejor}$  sustituye a  $\mathbf{y}_{3,1}$  para el individuo  $x_{3,2}$  en la siguiente iteración, debido a que tiene un mejor valor en la función objetivo y con esto cumple la condición correspondiente ( $f(\mathbf{y}_{3,1}) < f(\mathbf{p}_3^{mejor})$ ).

Por su parte, la Figura 3.30 (b), muestra la configuración de las partículas después de 15 iteraciones. Bajo esta configuración, los elementos de  $\mathbf{Y}_{15}$  tienen los siguientes valores en la función objetivo:  $f(\mathbf{y}_{1,15})=2.98$ ,  $f(\mathbf{y}_{2,15})=1.13$ ,  $f(\mathbf{y}_{3,15})=8.03$ ,  $f(\mathbf{y}_{4,15})=3.37$ ,  $f(\mathbf{y}_{5,15})=7.21$  y  $f(\mathbf{y}_{6,15})=0.84$ . Las partículas que han sido desplazadas a nuevas posición reducen el número de veces que se ejecuta la búsqueda local. Es evidente en la Figura 3.30 (b) que las partículas  $\mathbf{y}_{3,15}$  e  $\mathbf{y}_{5,15}$  están demasiado cerca la una de la otra y por lo tanto comparten una posición en los sub-espacios  $C_3$  y  $C_5$ . Debido a que el sub-espacio  $C_5$  contiene otro elemento ( $\mathbf{y}_{3,15}$ ) con un valor en la función objetivo mejor que el de  $\mathbf{y}_{5,15}$ , no cumple con las condiciones necesarias para que sea aplicada la BL. Por lo tanto, sólo el punto  $\mathbf{y}_{3,15}$  será procesado, generando dos muestras aleatorias dentro de  $C_3$ . Con el incremento de las iteraciones, las partículas tienden a concentrarse alrededor de una de las soluciones. En la Figura 3.30 (c) se muestra la configuración de las partículas después de 25 iteraciones. Es posible observar que todas las partículas están agrupadas alrededor del valor óptimo. Bajo estas circunstancias, el nuevo proceso de BL, solamente se ejecuta una vez, ya que la concentración de las partículas trabaja como un proceso de explotación, con varias partículas tratando de refinar su posición en el espacio de búsqueda, dentro del vecindario de las soluciones ya conocidas.



**Figura 3.30:** Funcionamiento del nuevo mecanismo de búsqueda local, (a) opera considerando la población inicial modificada  $\mathbf{Y}_1$ , (b) considerando población modificada  $\mathbf{Y}_{15}$  después de 15 iteraciones y (c) opera considerando la población modificada  $\mathbf{Y}_{25}$  después de 25 iteraciones.

### 3.9.2 Incorporación de la memoria

Como ya se mencionó previamente, la localización y reconocimiento de objetos mediante el uso de plantillas, tiene como propósito encontrar la mejor semejanza entre la plantilla  $R$  y la imagen  $I$ . Para esto, se define un espacio de búsqueda  $S$  que contiene un conjunto finito de posibles posiciones en que se puede localizar la plantilla. Sin embargo, ya que el algoritmo EMO emplea números aleatorios para el cálculo de los nuevos elementos de la población, las partículas de EMO pueden encontrar soluciones repetidas. Por ejemplo, posiciones que ya han sido visitadas por otros individuos en iteraciones previas. Este hecho restringe seriamente el rendimiento de EMO debido a que para este problema la evaluación de la correlación cruzada normalizada es computacionalmente costosa.

Para mejorar el rendimiento de la estrategia de búsqueda, el número de evaluaciones de NCC se reduce incorporando una memoria de afinidad (**FM**) para almacenar los valores de la función objetivo visitados previamente, con esto se evita reevaluar las mismas posiciones de las partículas. La memoria **FM** contiene una lista que incluye las posiciones de búsqueda y sus correspondientes valores de NCC. Por lo tanto, para una posición dada, se explora por adelantado la memoria **FM** para verificar si contiene tal posición, en caso afirmativo no se requiere una evaluación. En otro caso, se evalúa la posición en la función NCC y su valor es almacenado en **FM** para poder usarlo después. Este mecanismo de memoria puede aparentar ser una desventaja para imágenes de un gran tamaño, ya que se requiere almacenar una gran cantidad de valores. Sin embargo, ya que la memoria de afinidad sólo almacena los valores que ya han sido calculados, el número de localidades requeridas es menor que el tamaño de la imagen. Por lo tanto, es posible implementarla incluso en sistemas con recursos computacionales modestos.

## 3.10 Localización y detección de objetos mediante el uso de plantillas empleando EMO como estrategia de búsqueda

En esta sección se presenta la implementación del algoritmo EMO en conjunto con las modificaciones propuestas, como estrategia de búsqueda en la localización y detección de objetos mediante plantillas. Además, se muestran los resultados experimentales, tras aplicar el método propuesto a un conjunto de imágenes de prueba, asimismo se expone un

estudio comparativo del algoritmo propuesto con métodos que son referencia en la literatura.

### 3.10.1 Implementación

En el algoritmo propuesto, las partículas representan la posición de búsqueda definida por  $(u, v)$ , la cual se desplaza a través del espacio de búsqueda  $\mathbf{S}$ . Los coeficientes NCC se usan como valor de función objetivo, que evalúa la calidad de coincidencia presentada entre la imagen plantilla  $R$  y la imagen fuente  $I$ , para una determinada posición (individuo de la población). El número de evaluaciones de NCC se reduce considerando una versión mejorada del algoritmo EMO como estrategia de búsqueda. Por consiguiente, un conjunto de posiciones candidatas guiadas por los valores de la función objetivo, evolucionan usando los operadores de EMO que han sido modificados, esta evolución continua hasta que se encuentra la mejor semejanza posible.

El espacio de búsqueda  $\mathbf{S}$ , consiste en un conjunto de posiciones de búsqueda  $\hat{u}$  y  $\hat{v}$  bi-dimensionales que representan los componentes de cada localidad de búsqueda. Considerando que el tamaño de la imagen fuente  $I$  es  $M \times N$  y el tamaño de la plantilla  $R$  es  $m \times n$ , los límites del espacio de búsqueda  $\mathbf{S}$  se establecen por la diferencia de tamaños entre la imagen fuente y la plantilla. Para el límite inferior  $lb_1 = M - m$  y para el superior  $ub_2 = N - n$ . Por lo tanto, cada partícula es codificada como:

$$x_i = \{(\hat{u}_i, \hat{v}_i) \mid 1 \leq \hat{u}_i \leq lb_1, 1 \leq \hat{v}_i \leq ub_2\} \quad (3.33)$$

donde  $x_i^1$  y  $x_i^2$  representan las variables que serán modificadas dentro del proceso de optimización.

Considerando que cada partícula  $\mathbf{x}_i$  simboliza una localización  $(u, v)$  dentro del espacio de búsqueda  $\mathbf{S}$  y  $NCC(u, v)$  representa la calidad de similitud (función objetivo) para esta posición, el enfoque propuesto involucra el procedimiento que se describe a continuación. Primero, se inicializa la memoria de afinidad (**FM**), como un arreglo vacío. El siguiente paso, es crear una población de partículas, distribuidas aleatoriamente dentro del espacio de búsqueda. Entonces, todas las partículas son evaluadas en la función objetivo (NCC) y estos valores se almacenan en la respectiva posición de **FM**. Bajo el enfoque EMO, la carga de cada partícula se calcula de acuerdo al valor de NCC. Después, se calcula el vector total de fuerza y se actualiza la posición de la población. Cada vez que se va a

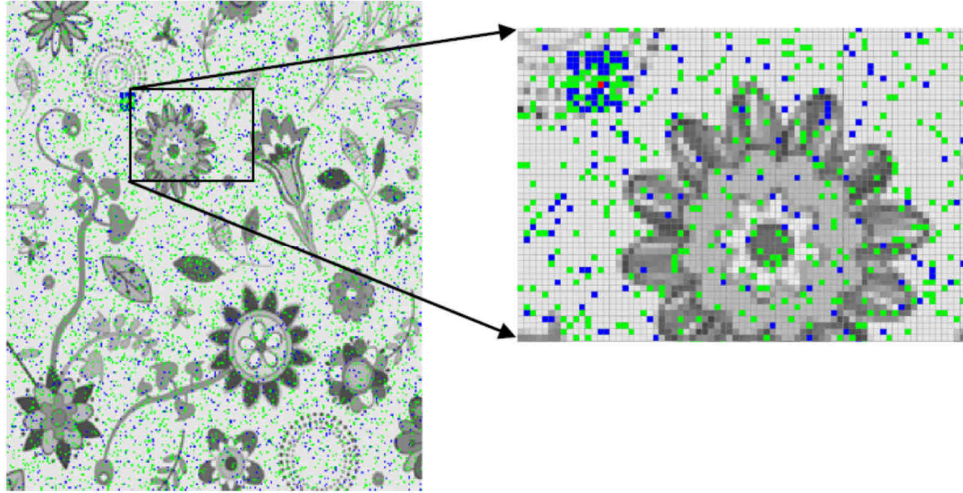
evaluar una nueva partícula, se debe verificar en **FM** si ha sido evaluada anteriormente. Si tal partícula existe en **FM**, no es necesario obtener su NCC, en caso contrario se evalúa y se almacena su valor. Una vez que las posiciones de las partículas son actualizadas, se aplica el nuevo algoritmo de búsqueda local, para mejorar la calidad de las soluciones. Este procedimiento se repite hasta que se alcanza un número de iteraciones previamente definido.

Por lo tanto, el algoritmo propuesto basado en EMO se puede resumir en los siguientes pasos:

- Paso 1:** Leer la imagen fuente en escala de grises  $I$ .
- Paso 2:** Seleccionar la plantilla  $R$ .
- Paso 3:** Inicializar la memoria de afinidad (**FM**), como un arreglo vacío.
- Paso 4:** Inicializa un conjunto  $\mathbf{X}_1$  de  $N$  partículas distribuidas en el espacio de búsqueda  $\mathbf{S}$  (Algoritmo 3.1).
- Paso 5:** Evaluar el coeficiente NCC para toda la población usando las ecuaciones (3.29) y (3.30) y almacenar los valores de NCC en **FM**.
- Paso 6:** Calcular la carga y el vector de fuerza total de las partículas usando el Algoritmo 3.3.
- Paso 7:** Mover las partículas hacia las nuevas posiciones basándose en la fuerza total (Algoritmo 3.4).
- Paso 8:** Antes de evaluar las nuevas posiciones de la población, se verifica en la memoria **FM** si tales posiciones ya han sido evaluadas.
- Paso 9:** Aplicar el nuevo procedimiento de búsqueda local (Algoritmo 3.5).
- Paso 10:** Seleccionar la mejor partícula de la población que tiene el valor de NCC más alto.
- Paso 11:** Si el número de iteraciones ha sido alcanzado, entonces determinar el mejor individuo de la población final, en otro caso regresar al paso 6.

El método propuesto considera múltiples localidades de búsqueda durante todo el proceso de optimización. Sin embargo, sólo unas pocas son evaluadas usando la función objetivo, las posiciones restantes se toman de la memoria **FM**. La Figura 3.31 muestra un segmento del patrón de búsqueda que ha sido generado por el algoritmo propuesto basado en EMO. La plantilla o patrón contiene las localidades de búsqueda evaluadas representadas con recuadros verdes, mientras que la localidad con el máximo valor de NCC está marcada



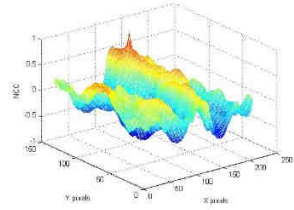


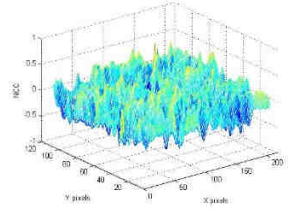
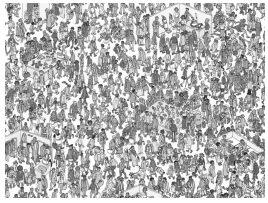

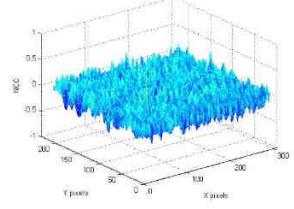


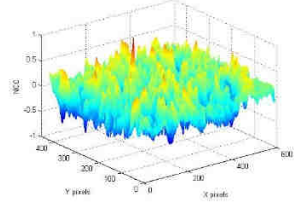


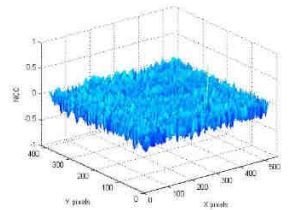
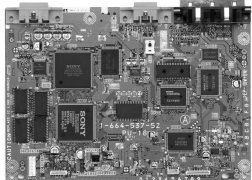

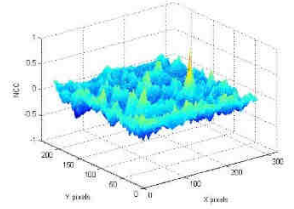
de color rojo en el cuadrante superior izquierdo. Por otro lado, las localidades de color azul representan los valores que han sido elegidos de forma repetida, las celdas que mantienen tonalidades en escala de gris no fueron visitadas durante el proceso de optimización.



**Figura 3.31:** Patrón de búsqueda generado por el algoritmo EMO, los puntos verdes representan las posiciones evaluadas, los puntos azules indican que los puntos ya han sido visitados y están almacenados en la **FM**. El punto rojo es el que tiene la similitud óptima.

### 3.10.2 Resultados experimentales

Para verificar la utilidad y efectividad del método propuesto, se han llevado a cabo un conjunto de experimentos comparativos. Dichos experimentos consideran un conjunto de imágenes que se muestran en la Tabla 3.20 con sus respectivas plantillas y tamaños. Para ilustrar la complejidad del proceso de localización y detección de objetos visto como un problema de optimización. La Tabla 3.20 también presenta las superficies de la función objetivo (NCC) producidas después de aplicar la búsqueda exhaustiva a cada imagen.

	Imagen	Plantilla a buscar	Superficie de NCC	Tamaño de la imagen (píxeles)
Perro				574×800
Soccer				720×399
Waldo				1024×768
Aeropuerto				2048×1536
Ciudad				550×413
PCB				3220×2400

**Tabla 3.20:** Algoritmo EMO para localización y detección de objetos mediante plantillas, aplicado a diferentes tipos de imágenes.

Los resultados obtenidos por el algoritmo propuesto se han comparado con los que se produjeron por trabajos similares reportados en la literatura referente, dichos métodos usan como estrategia de búsqueda los siguientes algoritmos: ICA (*Imperialist Competitive Algorithm*) [87], GA [84], PSO [86] y el EMO original [13]. Los primeros tres enfoques son considerados dentro del estado del arte de la presente memoria y sus resultados han sido publicados recientemente. Sin embargo, el algoritmo EMO original se ha incluido sólo para validar el rendimiento de la versión propuesta. El número máximo de iteraciones se configura a 300 como criterio de paro, el cual se ha elegido para mantener la compatibilidad con los trabajos usados para la comparación [84-87].

La configuración de los parámetros para cada algoritmo en las comparaciones se describe a continuación:

1. ICA:  $Num\_de\_Países = 100$ ,  $Num\_de\_Imper = 10$ ,  $Num\_de\_Colonias = 90$ ,  $T_{max} = 300$ ,  $\xi = 0.1$ ,  $\varepsilon_1 = 0.15$  y  $\varepsilon_2 = 0.9$ . Estos valores constituyen la mejor configuración de acuerdo a lo presentado en [87].
2. PSO:  $Tamaño\_enjambre = 50$ ,  $Peso\_inercia = 0.3925$ ,  $Mejor\_peso\_particula = 2.55$ ,  $Mejor\_peso\_enjambre = 1.33$  y  $Max\_iter = 300$ . De acuerdo a [185], estos valores representan la mejor configuración posible.
3. GA: Tamaño de la población es 70, la probabilidad de cruzamiento es 0.55, la probabilidad de mutación es 0.10, el número de mejores individuos (*élite*) es 2 y el número de generaciones es 300. También se aplican la selección por ruleta y el cruzamiento de un sólo punto [83].
4. EMO (original):  $numero\_partículas = 50$ ,  $\delta = 0.001$ ,  $LSITER = 4$  y  $MaxITER = 300$ . De acuerdo a las referencias [13, 18], estos valores representan la mejor configuración posible.
5. EMO (BL modificada):  $N = 50$ ,  $m = 5$ ,  $\beta = 0.05$  e  $ITER = 300$ . Estos valores se han encontrado por medio de la experimentación tras diversas pruebas de ensayo y error.

La configuración de los parámetros para todos los algoritmos no se modifica durante los experimentos. Las comparaciones se han analizado considerando cinco índices para medir el rendimiento de los métodos: el promedio del NCC (ANCC), el rango de éxito (SR), el promedio del número de localidades de búsqueda (ASL), el número de iteraciones (NI) y el promedio del tiempo computacional (CT). El promedio del NCC se calcula considerando el número total de ejecuciones de cada algoritmo. El rango de éxito, representa el porcentaje del número de ejecuciones en el cual los algoritmos encontraron



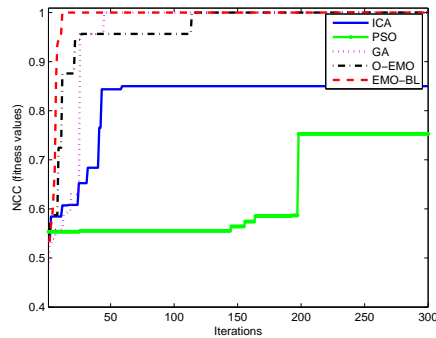
exitosamente el valor óptimo para la detección. El promedio del número de localidades de búsqueda, corresponde a la cantidad de localidades analizadas, las cuales han sido visitadas en un único experimento. Este índice de rendimiento se puede relacionar con el promedio de evaluaciones de la función NCC. El número de iteraciones indica en que iteración se encontró la mejor similitud. Finalmente, el tiempo computacional registra el tiempo en segundos que cada algoritmo utilizó para encontrar la mejor solución. Para garantizar una consistencia estadística todos los índices de rendimiento han sido calculados considerando 35 ejecuciones independientes para cada método.

Los resultados para las 35 ejecuciones se reportan en la Tabla 3.21 donde la mejor respuesta para cada imagen aparece en negrita. De acuerdo a esta tabla, el método propuesto proporciona mejores resultados para todas las imágenes en comparación con ICA, PSO, GA y EMO-original. En particular, esta prueba pone de manifiesto la gran diferencia en el rango de éxito (SR), el promedio del número de localidades de búsqueda (ASL), el tiempo computacional (CT) y el número de iteraciones (NI).

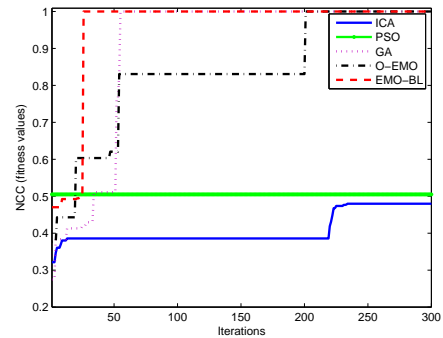
La Tabla 3.21 demuestra que el algoritmo EMO propuesto posee una mayor precisión en comparación con sus contrapartes, alcanzando los mejores valores promedio de NCC (cerca de 1). Igualmente, el algoritmo EMO con la búsqueda local modificada, presenta un mejor rendimiento en términos de efectividad, ya que logra detectar el punto óptimo en prácticamente todos los experimentos. Por otro lado, el método basado en EMO para detección de objetos, es capaz de reducir drásticamente el número de localidades de búsqueda (este valor es representado por la cantidad de veces que se evalúa el NCC), el tiempo computacional y el número de iteraciones. Es importante recordar que las evaluaciones de NCC representan el principal costo computacional asociado a esta técnica de detección de objetos. El rendimiento sobresaliente de la técnica propuesta se relaciona directamente con un mejor equilibrio entre la convergencia y el consumo de recursos de cómputo, esto es debido a la incorporación de la nueva BL y la memoria **FM**. Por su parte, la Figura 3.32 muestra la evolución de la curva de coincidencia para cada imagen, considerando el mejor valor promedio de NCC que se obtuvo para todos los algoritmos empleados en los experimentos. Las gráficas de dicha evolución se han calculado considerando una única ejecución de los algoritmos para las imágenes de la Tabla 3.20.

Imagen	Algoritmo	Valor NCC promedio (ANCC)	Rango de éxito (SR)%	Promedio de localidades de búsqueda (ASL)	Tiempo computacional (CT)	Numero de iteraciones (NI)
Perro	ICA	0.8856	70.21	29500	74.69	150
	PSO	0.6681	34.28	35680	73.43	180
	GA	0.5631	45.58	32697	101.23	172
	O-EMO	0.8281	90.54	46140	78.99	109
	EMO-BL	<b>1.0000</b>	<b>100</b>	<b>13800</b>	<b>34.92</b>	<b>72</b>
Soccer	ICA	0.5054	14.65	29621	10.09	165
	PSO	0.4050	2.85	35240	12.31	85
	GA	0.3753	22.98	30987	19.30	139
	O-EMO	0.7150	20.34	45000	11.96	144
	EMO-BL	<b>1.0000</b>	<b>100</b>	<b>16920</b>	<b>3.80</b>	<b>57</b>
Waldo	ICA	0.6587	60.43	28786	10.85	128
	PSO	0.2154	2.05	32169	10.54	99
	GA	0.2057	2156	31875	17.80	141
	O-EMO	0.6422	70.33	44775	12.36	188
	EMO-BL	<b>0.9598</b>	<b>98.00</b>	<b>16044</b>	<b>7.10</b>	<b>48</b>
Aeropuerto	ICA	0.6959	54.42	29177	55.23	67
	PSO	0.5655	2.85	28978	62.59	85
	GA	0.5676	28.56	25921	78.96	174
	O-EMO	0.9170	59.51	45580	61.38	142
	EMO-BL	<b>1.0000</b>	<b>100</b>	<b>17650</b>	<b>47.70</b>	<b>35</b>
Ciudad	ICA	0.2656	46.21	29399	8.84	61
	PSO	0.1777	2.00	31213	5.90	102
	GA	0.1583	20.56	30578	11.53	114
	O-EMO	0.6166	74.75	44651	9.21	127
	EMO-BL	<b>0.9843</b>	<b>97.00</b>	<b>15830</b>	<b>3.70</b>	<b>35</b>
PCB	ICA	0.3136	51.04	28985	69.38	135
	PSO	0.2090	2.00	30459	75.96	120
	GA	0.2015	18.26	36987	120.69	229
	O-EMO	0.6921	74.81	47689	89.74	195
	EMO-BL	<b>0.9067</b>	<b>92.65</b>	<b>16990</b>	<b>58.98</b>	<b>151</b>

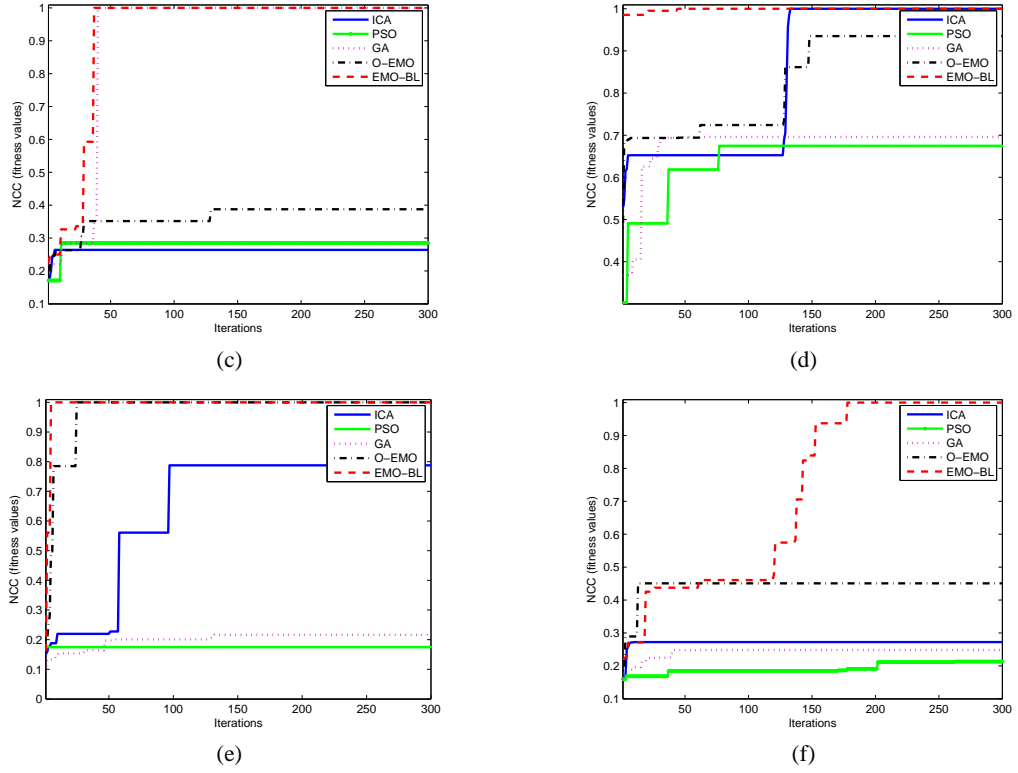
**Tabla 3.21:** Comparación del rendimiento de los método ICA, PSO, GA EMO y EMO con BL modificada para las imágenes experimentales presentadas en la Tabla 3.20.



(a)



(b)



**Figura 3.32:** Comparación de los valores de NCC para las imágenes: (a) perro, (b) soccer, (c) Waldo, (d) aeropuerto, (e) ciudad y (f) PCB, usando los cinco diferentes algoritmos.

De forma similar a otros métodos, se ha llevado a cabo la prueba de rangos de Wilcoxon para muestras independientes, para esta prueba se usó el número de localidades de búsqueda (ASL) de la Tabla 3.21 con un nivel de significancia del 5%. En la Tabla 3.22 se reportan los valores  $p$  producidos por la prueba de Wilcoxon para la comparación por pares de los valores ASL en cuatro grupos. Estos grupos están formados por EMO-BL contra ICA, EMO-BL contra PSO, EMO-BL contra GA, EMO-BL contra EMO-O. Como hipótesis nula se asume que no existe una diferencia significativa entre los algoritmos. La hipótesis alternativa considera una diferencia significativa entre los valores de ASL de los métodos comparados. Los valores mostrados en la Tabla 3.21 son menores que 0.05, lo cual constituye una fuerte evidencia de que la hipótesis nula no se cumple. Por lo tanto, se comprueba que los valores obtenidos por el método propuesto son estadísticamente significativos y no se presentan por una coincidencia.

Image	EMO-BL contra ICA	EMO-BL contra PSO	EMO-BL contra GA	EMO-BL contra O-EMO
Perro	6.5304e-13	6.4894e-08	7.0548e-13	6.5154e-13
Soccer	6.5335e-13	8.5675e-11	2.9973e-12	6.5094e-13
Waldo	6.5304e-13	5.7977e-12	8.3789e-13	6.5034e-13
Aeropuerto	6.5304e-13	4.2026e-12	9.5761e-12	6.5064e-13
Ciudad	6.5154e-13	1.6791e-11	7.7189e-13	6.4825e-13
PCB	6.5304e-13	8.5921e-12	4.3589e-12	6.4462e-13

**Tabla 3.22:** Valores  $p$ , producidos por la prueba de Wilcoxon comparando los valores ASL de la siguiente manera: EMO-BL contra ICA, EMO-BL contra PSO, EMO-BL contra GA y EMO-BL contra EMO-O.

### 3.11 Resumen

En este capítulo se ha presentado el algoritmo de optimización EMO como un método alternativo para dar solución a problemas de procesamiento de señales e imágenes digitales. Se han propuesto dos modificaciones al algoritmo EMO original con el propósito de mejorar su rendimiento, sin demeritar las características de exploración y explotación del espacio de búsqueda.

Para una de las modificaciones propuestas, se hizo uso de una regla de aprendizaje que está basada en la teoría de lo opuesto. Los resultados experimentales de dicha modificación, dan evidencia de las capacidades de esta nueva técnica para reducir el número de iteraciones requeridas para encontrar el valor óptimo global. Para la segunda modificación, se propone un nuevo proceso de búsqueda local, el cual tiene como propósito reducir el número de evaluaciones de la función objetivo. Para comprobar la efectividad de esta nueva propuesta se propone la implementación del algoritmo EMO modificado para la localización y detección de objetos mediante el uso de plantillas. Para esto también se introduce el uso de la memoria de afinidad, la cual, debido a las características del problema, permite reducir el tiempo de cómputo evitando así que las posibles soluciones sean re-evaluadas en la función objetivo.

Por otra parte, se propuso un nuevo algoritmo para la identificación y el modelado de filtros digitales, los cuales son ampliamente usados en procesamiento digital de señales. Este nuevo algoritmo propuesto incorpora el uso de EMO para estimar los parámetros de los filtros digitales. Los experimentos realizados demuestran que esta técnica es capaz de

modelar las plantas propuestas a partir de los filtros que han sido sintonizados usando EMO. Del mismo modo, a partir del detector de círculos basado en EMO se propone un detector de glóbulos blancos, que incorpora una nueva función objetivo en la cual se toman en cuenta unas métricas adicionales, que generan una mayor precisión en la detección de las células sanguíneas. Los experimentos realizados corroboran la exactitud con que los leucocitos son identificados, todo ello a pesar de las condiciones complejas que presentan las imágenes medicas seleccionadas.

Tanto para las modificaciones como para las aplicaciones de EMO, el estudio de los resultados experimentales incluye comparaciones con técnicas similares propuestas en la literatura y pruebas estadísticas no paramétricas que dan validez a los resultados obtenidos.

# Capítulo 4

## Algoritmo de optimización

### *Harmony Search:*

## segmentación multinivel de imágenes digitales

---

En este capítulo se implementa el algoritmo de optimización *Harmony Search* (HSA) para la segmentación multinivel de imágenes digitales, como primer punto se presenta la teoría referente a HSA donde son analizados los operadores empleados en el proceso de optimización. También se estudia la estructura básica de este optimizador y se resaltan las etapas clave del mismo.

Por otro lado, se presenta la teoría referente a la segmentación multinivel de imágenes digitales, a partir de aquí se deducen las funciones objetivo que son optimizadas por medio del algoritmo HSA para encontrar los mejores valores de umbral. Finalmente, se

muestran los resultados experimentales que evidencian de las capacidades de HSA para generar la solución del problema de segmentación multinivel.

#### 4.1 Algoritmo de optimización *Harmony Search* (HSA)

En el algoritmo *Harmony Search* básico cada solución es conocida como una “armonía”, siendo representada por un vector  $n$ -dimensional de números reales. Una población inicial de vectores de armonía son generados aleatoriamente y almacenados dentro de la memoria de armonías (*harmony memory*, HM). Una nueva armonía candidata es entonces generada a partir de los elementos de contenidos en la HM, para esto se usa un parámetro de consideración de la memoria, ya sea por inicialización aleatoria o por medio de una operación conocida como de ajuste de tono. Finalmente, la memoria de armonías es actualizada por medio de la comparación de la nueva armonía candidata y el peor de los vectores de armonía. El peor vector de armonía es remplazado por el nuevo vector candidato cuando éste sea considerado una mejor solución. El proceso anterior es repetido hasta que se alcanza un criterio de paro previamente establecido. El algoritmo HSA básico consta de tres fases principales: la inicialización de la HM, la improvisación de los nuevos vectores de armonías y la actualización de la HM. Cada una de estas fases será descrita en este capítulo.

*Inicialización del problema y de los parámetros del algoritmo:* como ya se mencionó en los capítulos anteriores, el problema de optimización global se puede resumir en lo siguiente:

$$\begin{aligned} \text{Minimizar} \quad & f(\mathbf{x}), \quad \mathbf{x} = [x(1), x(2), \dots, x(n)] \in \mathbb{R}^n, \\ \text{sueto a:} \quad & x(j) \in [l(j), u(j)] \quad j = 1, 2, \dots, n, \end{aligned} \tag{4.1}$$

donde  $f(\mathbf{x})$  es la función objetivo,  $\mathbf{x} = [x(1), x(2), \dots, x(n)]$ ,  $n$  es el número de variables de diseño,  $l(j)$  y  $u(j)$  son los límites inferior y superior para cada variable de diseño  $x(j)$  respectivamente. Los parámetros para el algoritmo HSA son: el tamaño de la memoria de armonías (HMS), la tasa o proporción en que es considerada la HM (HMCR), la proporción de ajuste de tono (PAR), la distancia de ancho de banda (BW) y el número de improvisaciones (NIm), que es igual al número total de iteraciones. El rendimiento del HSA está fuertemente influenciado por los valores asignados a estos parámetros, los cuales a su vez dependen de la aplicación del mismo [186].

*Inicialización de la memoria de armonías:* en esta etapa son configurados los vectores iniciales de la HM, el tamaño de la HM es definido como HMS. Se supone dado  $\mathbf{x}_i = [x_i(1), x_i(2), \dots, x_i(n)]$ , que representa el  $i$ -ésimo vector de armonías generado aleatoriamente por medio de la ecuación siguiente:

$$x_i(j) = l(j) + (u(j) - l(j)) \cdot \text{rand}(0,1) \quad (4.2)$$

donde  $j=1,2,\dots,n$  e  $i=1,2,\dots,HMS$ ,  $\text{rand}(0,1)$  es un número aleatorio uniformemente distribuido entre 0 y 1, los límites superior e inferior son definidos por  $u(j)$  e  $l(j)$ , respectivamente. Entonces la matriz que compone la HM se inicializa con una cantidad de vectores HMS como se muestra a continuación:

$$\mathbf{HM} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{HMS} \end{bmatrix} \quad (4.3)$$

*Improvisación de los nuevos vectores de armonía:* en esta fase, se construye un nuevo vector de armonía  $\mathbf{x}_{new}$  mediante la aplicación de tres operadores básicos de HSA: consideración de la memoria, re-inicialización aleatoria y ajuste de tono. La generación de una nueva armonía se conoce como el proceso de improvisación. En el paso denominado consideración de la memoria, el valor de la primera variable de decisión del nuevo vector de armonía  $x_{new}(1)$  es elegido aleatoriamente a partir del conjunto de valores que existen actualmente en la HM, en otras palabras, se elige aleatoriamente un elemento del conjunto  $[x_1(1), x_2(1), \dots, x_{HMS}(1)]$ . Para llevar a cabo esta operación, se genera un número aleatorio  $r_1$  dentro del rango  $[0,1]$ . Si el valor de  $r_1$  es menor que el parámetro de consideración de la memoria de armonía (HMCR), la variable de decisión  $x_{new}(1)$  se genera por medio de las consideraciones de la memoria, en otro caso,  $x_{new}(1)$  se obtiene a través de la re-inicialización aleatoria entre los límites del espacio de búsqueda. Los valores de las variables de decisión restantes  $x_{new}(2), x_{new}(3), \dots, x_{new}(n)$  son seleccionados mediante el mismo procedimiento. Este proceso corresponde a la exploración del espacio de búsqueda y por consiguiente, ambas operaciones, consideración de la memoria y re-inicialización aleatoria, pueden ser modeladas como se presenta a continuación.



$$x_{new}(j) = \begin{cases} x_i(j) \in [x_1(j), x_2(j), \dots, x_{HMS}(j)], & \text{con probabilidad } HMCR \\ l(j) + (u(j) - l(j)) \cdot rand(0,1), & \text{con probabilidad } 1 - HMCR \end{cases} \quad (4.4)$$

Cada componente obtenida por el proceso de consideración de la memoria se volverá a analizar para determinar si su tono debe ser ajustado. Para esta operación, el parámetro de ajuste de tono (PAR) es definido para asignar la frecuencia de ajuste y el ancho de banda (BW) para controlar la búsqueda local o explotación alrededor de los elementos seleccionados de la HM. Por lo tanto, la decisión de ajuste de tono se calcula como sigue:

$$x_{new}(j) = \begin{cases} x_{new}(j) = x_{new}(j) \pm rand(0,1) \cdot BW, & \text{con probabilidad } PAR \\ x_{new}(j), & \text{con probabilidad } 1 - PAR \end{cases} \quad (4.5)$$

El ajuste de tono es el responsable de generar nuevas armonías potenciales modificando ligeramente las posiciones de las variables respecto de las originales. Tal operación puede considerarse similar al proceso de mutación en los algoritmos evolutivos. Por lo tanto, la variable de decisión es perturbada ya sea por un número aleatorio distribuido entre [0,1] o por un BW o bien no ser modificado. Para proteger la operación de ajuste de tono, es importante asegurar que los puntos que se sitúan fuera del rango  $[l, u]$  deben ser reasignados, tomando los valores del máximo o mínimo de los límites, dependiendo de si su valor es menor o mayor que ellos.

*Actualización de la memoria de armonías:* después de que se genera un nuevo vector de armonía  $\mathbf{x}_{new}$ , la memoria de armonías es actualizada por el valor que prevalece de la competición o comparación entre  $\mathbf{x}_{new}$  y el peor vector de armonías  $\mathbf{x}_w$  en la HM. Por lo tanto  $\mathbf{x}_{new}$  puede reemplazar a  $\mathbf{x}_w$  y convertirse en un nuevo miembro de HM, en caso de que el valor de la función objetivo de  $\mathbf{x}_{new}$  es mejor que el de  $\mathbf{x}_w$ .

*Procedimiento computacional:* el procedimiento computacional del algoritmo HSA básico se puede resumir en los pasos presentados en el Algoritmo 4.1 [25]. El cual se ha implementado con fines de minimización, si la intención es maximizar la función objetivo, se requiere modificar el signo del paso 21:  $\mathbf{x}_w = \mathbf{x}_{new}$  if  $f(\mathbf{x}_{new}) > f(\mathbf{x}_w)$ . Para los propósitos de la implementación de este capítulo se usa la maximización.

**Algoritmo 4.1: Harmony Search básico**


---

```

1: Inicializar  $HMS$ ,  $HMCR$ ,  $PAR$ ,  $BW$ ,  $NI$ 
2: Inicializar HM
3: Calcular  $f(\mathbf{HM})$ 
4: while  $it \leq NI$  do
5:   for  $j = 1$  to  $n$  do
6:     if  $r_1 < HMCR$  then
7:        $x_{new}(j) = x_a(j)$  donde  $a \in (1, 2, \dots, HMS)$ 
8:       if  $r_2 < PAR$  then
9:          $x_{new}(j) = x_{new}(j) \pm r_3 \cdot BW$  donde  $r_1, r_2, r_3 \in rand(0,1)$ 
10:      end if
11:      if  $x_{new}(j) < l(j)$  then
12:         $x_{new}(j) = l(j)$ 
13:      end if
14:      if  $x_{new}(j) > u(j)$  then
15:         $x_{new}(j) = u(j)$ 
16:      end if
17:    else
18:       $x_{new}(j) = l(j) + r \cdot (u(j) - l(j))$  donde  $r \in rand(0,1)$ 
19:    end if
20:  end for
21:  if  $f(\mathbf{x}_{new}) < f(\mathbf{x}_w)$  else
22:     $\mathbf{x}_w = \mathbf{x}_{new}$ 
23:  end if
24: end while

```

---

## 4.2 Segmentación multinivel de imágenes digitales

Como se mencionó en capítulos precedentes, el proceso más sencillo para segmentar una imagen en diferentes niveles de intensidad es la umbralización o umbralado. En este proceso los píxeles en escala de grises son divididos en conjuntos o clases dependiendo de su nivel de intensidad ( $L$ ). Para llevar a cabo esta clasificación, es necesario elegir correctamente un valor de umbral (*threshold*,  $th$ ) y simplemente considerar la siguiente regla:

$$\begin{aligned} C_1 &\leftarrow p \quad \text{si} \quad 0 \leq p < th, \\ C_2 &\leftarrow p \quad \text{si} \quad th \leq p < L-1 \end{aligned} \quad (4.6)$$

donde  $p$  es uno de los  $m \times n$  píxeles pertenecientes a una imagen en escala de grises  $I_g$ , la cual puede ser representada en  $L$  niveles de intensidad  $L = \{0, 1, 2, \dots, L-1\}$ .  $C_1$  y  $C_2$  son las clases a las que el píxel  $p$  puede pertenecer, mientras que  $th$  es el valor de umbral. La regla dada en la ecuación (4.6) corresponde a la umbralización bi-nivel y puede extenderse fácilmente para múltiples conjuntos o clases, tal como se muestra en la ecuación (4.7).

$$\begin{aligned}
C_1 &\leftarrow p & \text{si} & 0 \leq p < th_1, \\
C_2 &\leftarrow p & \text{si} & th_1 \leq p < th_2, \\
C_i &\leftarrow p & \text{si} & th_i \leq p < th_{i+1}, \\
&\vdots & & \vdots \\
C_n &\leftarrow p & \text{si} & th_n \leq p < L-1
\end{aligned} \tag{4.7}$$

donde  $\{th_1, th_2, \dots, th_i, th_{i+1}, th_k\}$  representa los diferentes valores de umbral para cada clase.

El problema tanto para la segmentación bi-nivel como multinivel consiste en seleccionar correctamente los valores de umbral que definan bien las clases. Aunque los métodos de Otsu y Kapur son enfoques conocidos para determinar tales valores, ambos proponen diferentes funciones objetivo, las cuales deben ser maximizadas con el fin de encontrar los valores de umbral óptimo. En las próximas sub-secciones se estudian ambas técnicas.

#### 4.2.1 Varianza entre clases (método de Otsu)

Esta es una técnica de umbralado no paramétrica, la cual fue propuesta por Otsu [64], emplea como criterio de segmentación, el máximo valor de varianza de las diferentes clases en las cuales serán divididos los píxeles. Para realizar este proceso, se toman los  $L$  niveles de intensidad de una imagen en escala de grises o de cada plano de una imagen de color RGB (*red*, *green* y *blue*; rojo, verde y azul). La distribución de probabilidad de los valores de intensidad se calcula como se define a continuación:

$$Ph_i^c = \frac{h_i^c}{NP}, \quad \sum_{i=1}^{NP} Ph_i^c = 1 \tag{4.8}$$

$$c = \begin{cases} 1, 2, 3 & \text{si la imagen es RGB} \\ 1 & \text{si la imagen es a escala de grises} \end{cases}$$

donde  $i$  es un nivel de intensidad específico ( $0 \leq i \leq L-1$ ),  $c$  es la componente de la imagen, la cual varía dependiendo de si se trata de una imagen en escala de grises o RGB, mientras que  $NP$  es la cantidad total de píxeles;  $h_i^c$  corresponde al histograma de la imagen, exactamente, representa el número de píxeles que corresponden al nivel de intensidad  $i$  en la componente  $c$ . El histograma es normalizado dentro de una distribución de probabilidad  $Ph_i^c$ . Para el tipo de segmentación más simple (bi-nivel) se definen dos clases:

$$C_1 = \frac{Ph_1^c}{\omega_0^c(th)}, \dots, \frac{Ph_{th}^c}{\omega_0^c(th)}, \quad C_2 = \frac{Ph_{th+1}^c}{\omega_1^c(th)}, \dots, \frac{Ph_L^c}{\omega_1^c(th)} \tag{4.9}$$

donde  $\omega_0(th)$  y  $\omega_1(th)$  son las distribuciones de probabilidades para  $C_1$  y  $C_2$ , como muestra la ecuación (4.10).

$$\omega_0^c(th) = \sum_{i=1}^{th} Ph_i^c, \quad \omega_1^c(th) = \sum_{i=th+1}^L Ph_i^c \quad (4.10)$$

Es necesario calcular la media de los niveles  $\mu_0^c$  y  $\mu_1^c$  que definen las clases, para esto se usa la ecuación (4.11). Una vez calculados estos valores, la varianza entre clases de Otsu  $\sigma^{2^c}$  es obtenida por medio de la ecuación (4.12).

$$\mu_0^c = \sum_{i=1}^{th} \frac{iPh_i^c}{\omega_0^c(th)}, \quad \mu_1^c = \sum_{i=th+1}^L \frac{iPh_i^c}{\omega_1^c(th)}, \quad (4.11)$$

$$\sigma^{2^c} = \sigma_1^c + \sigma_2^c \quad (4.12)$$

Debe tenerse en cuenta que para ambas ecuaciones (4.11) y (4.12),  $c$  depende del tipo de imagen, en la ecuación (4.12) el superíndice dos es parte del operador de varianza de Otsu y no representa ningún exponente en el sentido matemático. Además,  $\sigma_1^c$  y  $\sigma_2^c$  son las varianzas de las clases  $C_1$  y  $C_2$  las cuales se definen como:

$$\sigma_1^c = \omega_0^c (\mu_0^c + \mu_T^c)^2, \quad \sigma_2^c = \omega_1^c (\mu_1^c + \mu_T^c)^2 \quad (4.13)$$

donde  $\mu_T^c = \omega_0^c \mu_0^c + \omega_1^c \mu_1^c$  y  $\omega_0^c + \omega_1^c = 1$ . En base a los valores de  $\sigma_1^c$  y  $\sigma_2^c$  es posible definir la siguiente función objetivo:

$$J(th) = \max(\sigma^{2^c}(th)), \quad 0 \leq th \leq L-1 \quad (4.14)$$

Según lo anterior, el problema de optimización se reduce a encontrar el nivel de intensidad  $(th)$  que maximice la ecuación (4.14).

El método de Otsu es aplicado para una componente simple de una imagen, es por esto que para las imágenes RGB se aplicará para cada uno de los canales de la imagen. La descripción previa de esta técnica, a pesar de ser para un problema bi-nivel, puede extenderse para obtener múltiples umbrales. Considerando  $k$  umbrales, es posible separar la imagen original en  $k$  clases usando la ecuación (4.7), entonces es necesario calcular  $k$  varianzas y sus respectivos elementos. La función objetivo  $J(th)$  de la ecuación (4.14) puede entonces ser reescrita para emplearse con múltiples umbrales, como se explica a continuación:

$$J(\mathbf{TH}) = \max(\sigma^{2^c}(\mathbf{TH})), \quad 0 \leq th_i \leq L-1, \quad i=1,2,\dots,k \quad (4.15)$$

donde  $\mathbf{TH} = [th_1, th_2, \dots, th_{k-1}]$  es un vector que contiene los múltiples umbrales, de forma que las varianzas se calculan por medio de la ecuación (4.16).

$$\sigma^{2^c} = \sum_{i=1}^k \sigma_i^c = \sum_{i=1}^k \omega_i^c (\mu_i^c - \mu_T^c)^2 \quad (4.16)$$

Aquí,  $i$  representa el número de clase que se está analizando,  $\omega_i^c$  y  $\mu_i^c$  son respectivamente, la probabilidad de ocurrencia y la media de la clase. En términos de múltiples umbrales, tales valores son obtenidos como se describe a continuación:

$$\begin{aligned} \omega_0^c(th) &= \sum_{i=1}^{th_1} Ph_i^c, \\ \omega_1^c(th) &= \sum_{i=th_1+1}^{th_2} Ph_i^c, \\ &\vdots \\ \omega_{k-1}^c(th) &= \sum_{i=th_{k-1}+1}^L Ph_i^c, \end{aligned} \quad (4.17)$$

y los valores de la media de cada clase se calculan de la siguiente manera:

$$\begin{aligned} \mu_0^c &= \sum_{i=1}^{th_1} \frac{i Ph_i^c}{\omega_0^c(th_1)}, \\ \mu_1^c &= \sum_{i=th_1+1}^{th_2} \frac{i Ph_i^c}{\omega_1^c(th_2)}, \\ &\vdots \\ \mu_{k-1}^c &= \sum_{i=th_{k-1}+1}^L \frac{i Ph_i^c}{\omega_{k-1}^c(th_k)}, \end{aligned} \quad (4.18)$$

De forma similar al caso bi-nivel, para el umbralado multinivel usando el método de Otsu,  $c$  corresponde a los componentes de la imagen, para RGB  $c = 1, 2, 3$  y para escala de grises  $c = 1$ .

#### 4.2.2 Criterio de la entropía (método de Kapur)

Otro método no paramétrico que se usa para determinar los valores de umbral óptimo fue propuesto por Kapur y sus colaboradores [65]. Está basado en la entropía y en la distribución de probabilidad del histograma de la imagen. Esta técnica tiene como objetivo encontrar el  $th$  óptimo que maximice la entropía global de las clases. La entropía de una imagen se usa para medir la compactación y separabilidad que existe entre las clases. En este sentido, cuando un valor de umbral óptimo separa las clases, la entropía toma su valor

máximo. La función objetivo de Kapur para un ejemplo de segmentación bi-nivel puede definirse como:

$$J(th) = H_1^c + H_2^c, \quad (4.19)$$

$$c = \begin{cases} 1, 2, 3 & \text{si la imagen es RGB} \\ 1 & \text{si la imagen es a escala de grises} \end{cases}$$

donde las entropías  $H_1^c$  y  $H_2^c$  son calculadas de acuerdo al siguiente modelo:

$$H_1^c = \sum_{i=1}^{th} \frac{Ph_i^c}{\omega_0^c} \ln \left( \frac{Ph_i^c}{\omega_0^c} \right), \quad H_2^c = \sum_{i=th+1}^L \frac{Ph_i^c}{\omega_1^c} \ln \left( \frac{Ph_i^c}{\omega_1^c} \right) \quad (4.20)$$

$Ph_i^c$  es la distribución de probabilidad de los niveles de intensidad y es obtenida usando la ecuación (4.8).  $\omega_0^c(th)$  y  $\omega_1^c(th)$  son las distribuciones de probabilidad para  $C_1$  y  $C_2$ , mientras que  $\ln(\cdot)$  corresponde al logaritmo natural. De forma similar al método de Otsu, el enfoque basado en entropía puede extenderse a múltiples valores de umbral, para este caso es necesario dividir la imagen en  $k$  clases usando una cantidad similar de umbrales. Bajo estas condiciones, la nueva función objetivo se define como:

$$J(\mathbf{TH}) = \max \left( \sum_{i=1}^k H_i^c \right), \quad (4.21)$$

$$c = \begin{cases} 1, 2, 3 & \text{si la imagen es RGB} \\ 1 & \text{si la imagen es a escala de grises} \end{cases}$$

donde  $\mathbf{TH} = [th_1, th_2, \dots, th_{k-1}]$  es un vector que contiene los múltiples umbrales. Cada entropía se calcula por separado de acuerdo a su respectivo valor de umbral. De esa forma la ecuación (4.20) puede expandirse para  $k$  entropías.

$$\begin{aligned} H_1^c &= \sum_{i=1}^{th_1} \frac{Ph_i^c}{\omega_0^c} \ln \left( \frac{Ph_i^c}{\omega_0^c} \right), \\ H_2^c &= \sum_{i=th_1+1}^{th_2} \frac{Ph_i^c}{\omega_1^c} \ln \left( \frac{Ph_i^c}{\omega_1^c} \right), \\ &\vdots \\ H_k^c &= \sum_{i=th_{k-1}+1}^L \frac{Ph_i^c}{\omega_{k-1}^c} \ln \left( \frac{Ph_i^c}{\omega_{k-1}^c} \right) \end{aligned} \quad (4.22)$$

Los valores de la probabilidad de ocurrencia ( $\omega_0^c, \omega_1^c, \dots, \omega_{k-1}^c$ ) de las  $k$  clases son obtenidos usando la ecuación (4.17). Finalmente, es necesario usar la ecuación (4.7) para separar los píxeles en las clases correspondientes identificadas por los umbrales.

### 4.3 Segmentación multinivel de imágenes digitales usando *Harmony Search* (MTHSA)

En esta sección, se describe la implementación del algoritmo de optimización *Harmony Search* para encontrar los mejores umbrales que solucionen el problema de segmentación multinivel. Bajo el enfoque de HSA, primeramente se describe la representación de las armonías, a continuación se presenta la implementación del algoritmo MTHSA, finalmente se muestra la configuración de los parámetros de HSA para obtener los resultados óptimos según el problema planteado.

#### 4.3.1 Representación de las armonías

Cada armonía es una solución candidata que usa  $k$  diferentes elementos como variables de decisión en el algoritmo de optimización. Estas variables de decisión representan un punto diferente de umbral  $th$  que se usa para la segmentación. Por lo tanto, la población completa de armonías y umbrales se representa como:

$$\mathbf{HM} = \left[ \mathbf{x}_1^c, \mathbf{x}_2^c, \dots, \mathbf{x}_{HMS}^c \right]^T, \quad (4.23)$$

$$\mathbf{x}_i^c = \left[ th_1^c, th_2^c, \dots, th_k^c \right]$$

donde  $T$  se refiere al operador transpuesto de un vector,  $HMS$  es el tamaño de la memoria de armonías,  $\mathbf{x}_i^c$  es el  $i$ -ésimo elemento de la  $HM$  y  $c = 1, 2, 3$  para una imagen RGB mientras que  $c = 1$  para una imagen en escala de grises. Para este problema, los límites del espacio de búsqueda se configuran para  $l = 0$  y  $u = 255$ , que corresponden a los niveles de intensidad de la imagen.

#### 4.3.2 Implementación del algoritmo MTHSA

El algoritmo de segmentación propuesto ha sido implementando considerando dos funciones objetivo: la de Otsu y la de Kapur. Por lo tanto, el algoritmo HSA ha sido acoplado con ambas funciones produciendo dos diferentes algoritmos de segmentación. La implementación de ambas técnicas puede resumirse en el Algoritmo 4.2.

**Algoritmo 4.2: Harmony Search para múltiples umbrales (MTHSA)**


---

```

1: Leer la imagen  $I$ 
2: if  $I == RGB$  then
3:     Separar  $I$  en  $I_R, I_G, I_B$ ,  $c=1,2,3$ 
4:     Obtener histogramas  $h^R, h^G, h^B$ 
5: else
6:      $I_{GR}$  y  $c=1$ 
7:     Obtener el histograma  $h^{GR}$ 
8: end if
9: Calcular la distribución de probabilidad  $Ph_i^c$  usando la ecuación (4.8)
10: Inicializar  $HMS, HMCR, PAR, BW, NIm$ 
11: Inicializar HM
12: Calcular los valores de  $\omega_i^c$  y  $\mu_i^c$ 
13: Calcular  $f(\mathbf{HM})$ 
14: while  $it \leq NI$  do
15:     for  $j=1$  to  $n$  do
16:         if  $r_1 < HCMR$  then
17:              $x_{new}^c(j) = x_a^c(j)$  donde  $a \in (1, 2, \dots, HMS)$ 
18:             if  $r_2 < PAR$  then
19:                  $x_{new}^c(j) = x_{new}^c(j) \pm r_3 \cdot BW$  donde  $r_1, r_2, r_3 \in rand(0,1)$ 
20:             end if
21:             if  $x_{new}^c(j) < l(j)$  then
22:                  $x_{new}^c(j) = l(j)$ 
23:             end if
24:             if  $x_{new}^c(j) > u(j)$  then
25:                  $x_{new}^c(j) = u(j)$ 
26:             end if
27:         else
28:              $x_{new}^c(j) = l(j) + r \cdot (u(j) - l(j))$  donde  $r \in rand(0,1)$ 
29:         end if
30:     end for
31:     if  $f(\mathbf{x}_{new}^c) < f(\mathbf{x}_w^c)$  else
32:          $\mathbf{x}_w^c = \mathbf{x}_{new}^c$ 
33:     end if
34: end while
35: Seleccionar la mejor armonía  $\mathbf{x}_{best}^c$  de  $\mathbf{HM}$ 
36: Aplicar los umbrales de  $\mathbf{x}_{best}^c$  en la imagen usando la ecuación (4.7)

```

---

### 4.3.3 Configuración de los parámetros de MTHSA

El rendimiento de HSA está fuertemente influenciado por los valores asignados a los parámetros HM, HMCR, PAR, BW y NIm. Determinar los valores más apropiados de dichos parámetros para un problema arbitrario es una tarea compleja, ya que los parámetros interactúan entre ellos de una manera altamente no lineal y actualmente no existen modelos matemáticos que describan tal interacción. El método común para encontrar el mejor



conjunto de valores para los parámetros es asignar a cada uno de ellos un número aleatorio entre los límites de dicho parámetro y después ejecutar el HSA. Si el resultado final no es satisfactorio, entonces se define un nuevo conjunto de parámetros y el algoritmo se ejecuta de nuevo. Es evidente que este proceso puede resultar muy costoso en términos computacionales, ya que requiere una gran cantidad de intentos para encontrar un conjunto de parámetros adecuado. Con el propósito de reducir la cantidad de experimentos, en esta investigación se ha usado el método de diseño factorial [187, 188], para que de forma sistemática sean elegidos los parámetros del algoritmo HSA.

El método de diseño factorial [189] es una técnica estadística que evalúa al mismo tiempo todas las variables del proceso para determinar cuáles realmente influyen en el resultado final. Todas estas variables son llamadas factores y los diferentes valores elegidos para estudiar tales factores se denominan niveles. Los factores considerados en el diseño factorial son los parámetros del algoritmo HSA y el valor de la función objetivo es considerada como la respuesta. La Tabla 4.1 muestra los niveles de los valores cuantitativos usados en el diseño factorial. Los valores del nivel cero (conocido como punto central), se basan en lo que la literatura relacionada sugiere [188].

<b>Parámetros HSA (Factores)</b>	<b>Nivel (-)</b>	<b>Punto central</b>	<b>Nivel (+)</b>
HM	50	100	200
HMCR	0.5	0.75	0.9
PAR	0.1	0.5	0.9
BW	0.5	0.3	0.5
NIm	200	250	300

**Tabla 4.1:** Niveles de los factores empleados para el método de diseño factorial.

Cada experimento es llevado a cabo combinando dos niveles diferentes que definen cada parámetro, para esto se considera como problema el histograma de una imagen. Ya que se manejan cinco factores, se elige un diseño factorial fraccional de  $2^{5-1}$ , que requiere dieciséis experimentos de optimización más una prueba adicional para el punto central. Los resultados obtenidos tras las diecisiete ejecuciones han sido analizados usando una forma lineal de varianza conocida como ANOVA [189], se considera un 95% de confiabilidad. Después de cada análisis, se ha encontrado la mejor configuración posible de los

parámetros de HSA, los cuales se muestran en la Tabla 4.2. Estos resultados son consistentes, considerando seis repeticiones en las que se usaron diferentes histogramas y las funciones de Otsu y Kapur de forma indistinta.

HM	HMCR	PAR	BW	NIm
100	0.75	0.5	0.5	300

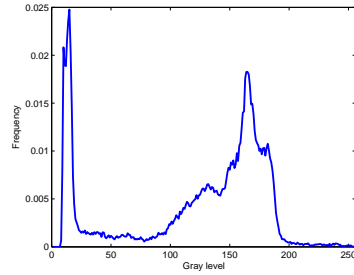
**Tabla 4.2:** Parámetros de HSA obtenidos por el método de diseño factorial.

#### 4.3.4 Resultados experimentales para la segmentación multinivel usando HSA

El algoritmo de segmentación multinivel basado en HSA ha sido probado usando un conjunto de once imágenes de referencia. Algunas de estas imágenes, son ampliamente usadas en la literatura referente al procesamiento de imágenes para probar diferentes métodos [62, 110]. Algunos ejemplos de estas imágenes son: *Lena*, *Camerman*, *Hunter*, *Baboon*, entre otros. Todas las imágenes tienen el mismo tamaño ( $512 \times 512$  píxeles) y están en formato JPEG. Para fines demostrativos, sólo cinco imágenes son usadas para representar gráficamente los resultados de la técnica propuesta. Sin embargo, para el análisis de los resultados numéricos, sí se emplea completamente el conjunto de prueba. Las cinco imágenes y sus respectivos histogramas se muestran en la Figura 4.1.



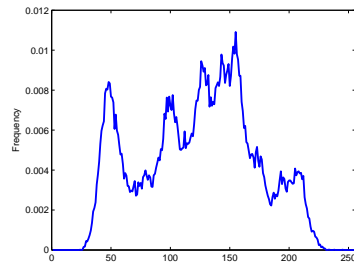
(a)



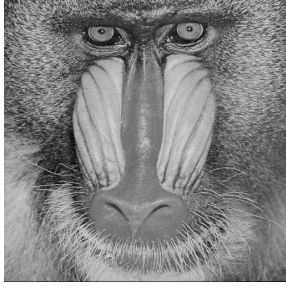
(b)



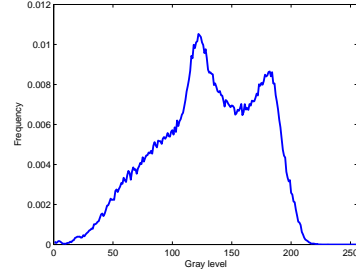
(c)



(d)



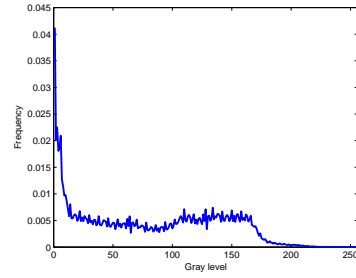
(e)



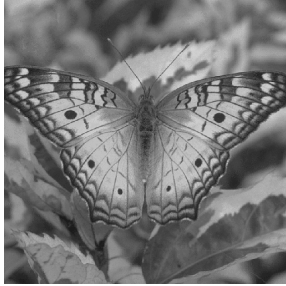
(f)



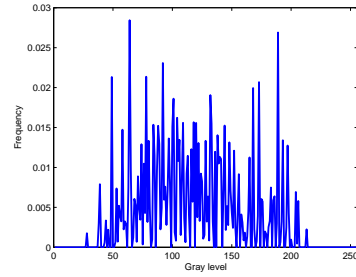
(g)



(h)



(i)



(j)

**Figura 4.1:** (a) *Cameraman*, (c) *Lena*, (e) *Baboon*, (g) *Hunter* y (i) *Butterfly*, imágenes seleccionadas. (b), (d), (f), (h) y (j) histogramas correspondientes a cada imagen.

Ya que el algoritmo HSA es estocástico, es necesario emplear métricas estadísticas que sean apropiadas para medir su eficiencia. Por lo tanto, los resultados se reportan ejecutando el algoritmo 35 veces para cada imagen. Para mantener la compatibilidad con los trabajos similares reportados en la literatura [190, 191, 36, 70], las cantidades de umbrales a encontrar son  $th = 2, 3, 4, 5$  para cada una de las imágenes. Por otro lado, el criterio de parada en los experimentos es el número de veces que se repite el mejor valor de la función objetivo. Por lo tanto, si el valor de la función objetivo para la mejor armonía se mantiene sin cambios durante el 10% del número total de iteración (NIm), entonces el algoritmo HSA se detendrá. Para evaluar la estabilidad y consistencias se calcula la desviación estándar (STD) de los resultados obtenidos en las 35 veces que se ejecutó el

algoritmo. La STD es una medida que expresa la dispersión de los datos, por lo tanto, si dicho valor se incrementa, entonces el algoritmo será más inestable [70]. La ecuación (4.24) muestra el modelo usado para calcular los valores de STD:

$$STD = \sqrt{\sum_{i=1}^{NI} \frac{bf_i - av}{Ru}} \quad (4.24)$$

donde  $bf_i$  es el mejor valor de la función objetivo en la  $i$ -ésima iteración,  $av$  es el promedio del valor de  $bf$  y  $Ru$  es el número total de veces que se ejecuta el algoritmo ( $Ru = 35$ ).

Por otro lado, se usa como índice de calidad la relación señal a ruido de pico (*peak-to-signal ratio*, PSNR), el cual mide la similitud entre la imagen segmentada y la imagen original. La imagen se segmenta usando los umbrales encontrados por HSA. Para realizar el cálculo del PSNR es necesario obtener el error cuadrático medio (RMSE) [68, 192], ambas métricas son definidas como:

$$PSNR = 20 \log_{10} \left( \frac{255}{RMSE} \right), \quad (dB)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{ro} \sum_{j=1}^{co} (I_o^c(i, j) - I_{th}^c(i, j))^2}{ro \times co}} \quad (4.25)$$

donde  $I_o^c$  es la imagen original,  $I_{th}^c$  es la imagen segmentada,  $c=1$  para imágenes en escala de grises y  $c=1,2,3$  para imágenes RGB, mientras que  $ro$  y  $co$  son el número total de filas y columnas que tiene la imagen.

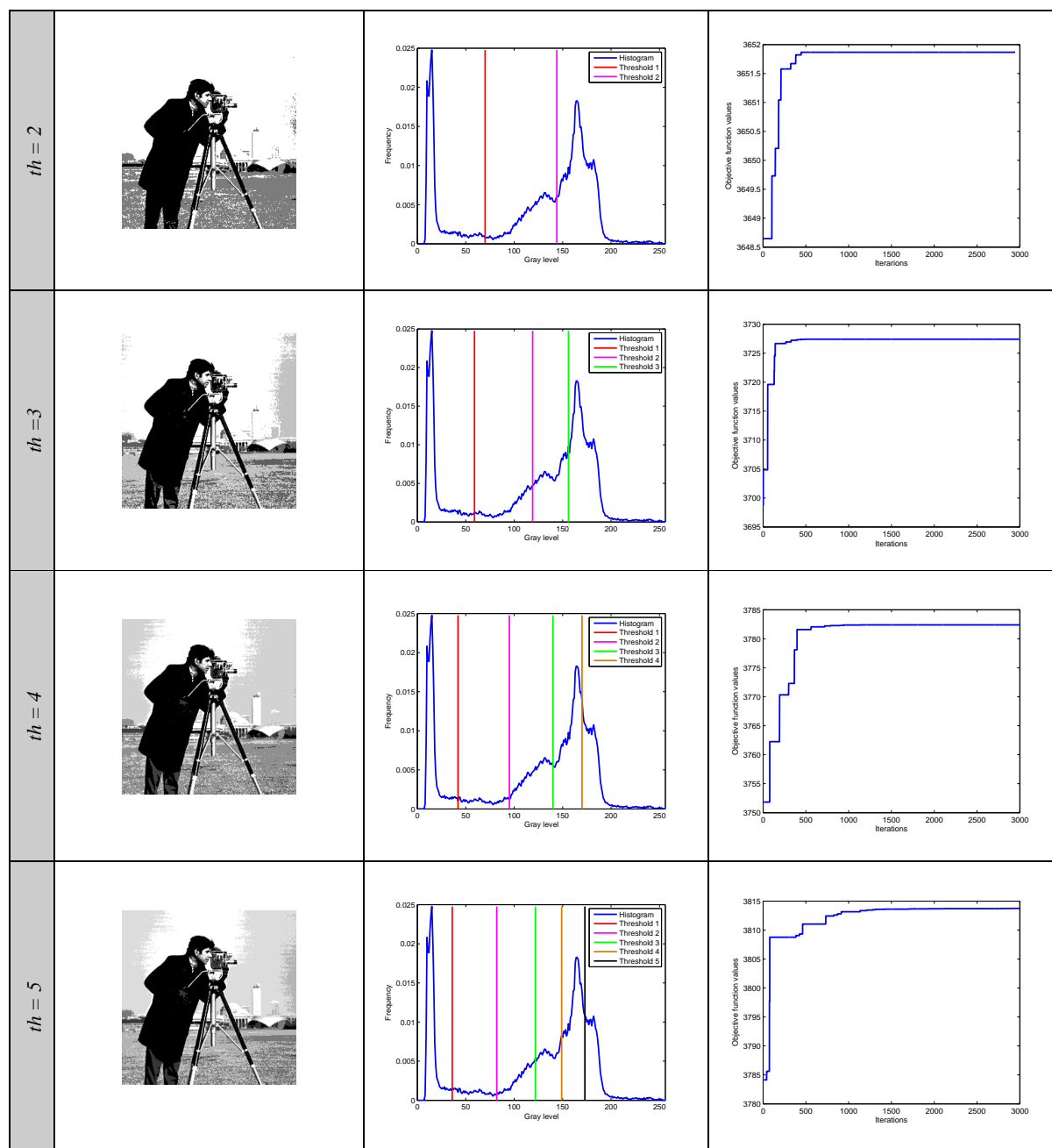
#### 4.3.4.1 Resultados de HSA usando la ecuación de Otsu

Aquí se analizan los resultados del algoritmo HSA después de considerar como función objetivo la varianza entre clases presentada en la ecuación (4.15), que se conoce ampliamente como el método de Otsu [64]. La técnica propuesta es aplicada a todo el conjunto de imágenes de prueba, los resultados se registran en la Tabla 4.3. En los resultados se presentan los mejores valores de umbral tras haber probado el algoritmo con cuatro diferentes valores de umbrales a detectar  $th = 2, 3, 4, 5$ . En esta tabla también se incluyen los valores PSNR y STD, resulta evidente que tales métricas incrementan su valor debido a que la cantidad de puntos umbral también se incrementa. Como se mencionó anteriormente, sólo son consideradas cinco imágenes para mostrar gráficamente los resultados de la segmentación. Tales imágenes cuentan con histogramas irregulares, bajo


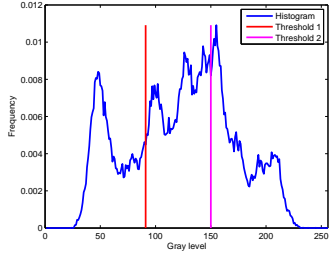
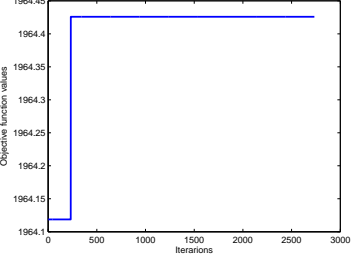

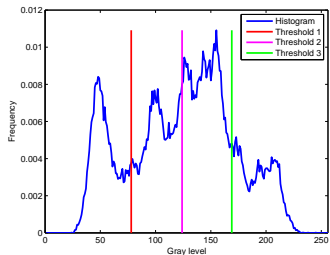
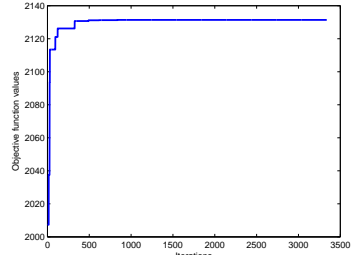

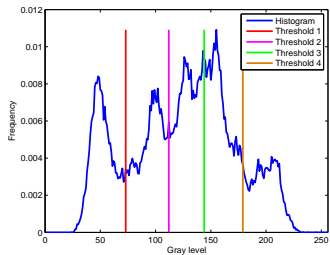
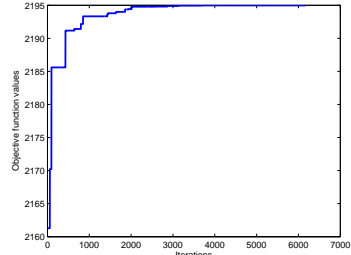

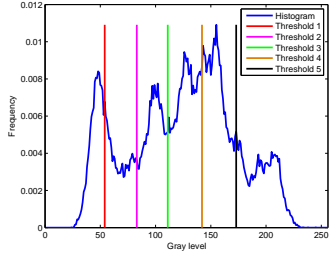
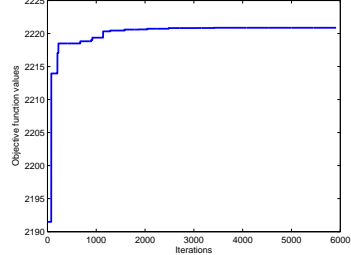
estas circunstancias, los métodos clásicos tienen grandes dificultades para encontrar los mejores valores de umbral. Los resultados aplicados a las imágenes seleccionados se presentan en cinco tablas: Tabla 4.4, 4.5, 4.6, 4.7 y 4.8; además de las imágenes sementadas, también se muestra la evolución del valor de la función objetivo para cada uno de los casos.

Imagen	$k$	Umbrales $x_{best}^c$	PSNR	STD
Cameraman	2	70, 144	17.247	2.30 E-12
	3	59, 119, 156	20.211	1.55 E-02
	4	42, 95, 140, 170	21.533	2.76 E-12
	5	36, 82, 122, 149, 173	23.282	5.30 E-03
Lena	2	91, 150	15.401	9.22 E-13
	3	79, 125, 170	17.427	2.99 E-02
	4	73, 112, 144, 179	18.763	2.77 E-01
	5	71, 107, 134, 158, 186	19.443	3.04 E-01
Baboon	2	97, 149	15.422	6.92 E-13
	3	85, 125, 161	17.709	1.92 E-02
	4	71, 105, 136, 167	20.289	5.82 E-02
	5	66, 97, 123, 147, 173	21.713	4.40 E-01
Hunter	2	51, 116	17.875	2.30 E-12
	3	36, 86, 135	20.350	2.30 E-12
	4	27, 65, 104, 143	22.203	1.22 E-02
	5	22, 53, 88, 112, 152	23.703	1.84 E-12
Airplane	2	113, 173	15.029	9.22 E-13
	3	92, 144, 190	18.854	4.83 E-01
	4	84, 129, 172, 203	20.735	7.24 E-01
	5	68, 106, 143, 180, 205	23.126	8.38 E-01
Peppers	2	72 138	16.299	1.38 E-12
	3	65 122 169	18.359	4.61 E-13
	4	50 88 128 171	20.737	4.61 E-13
	5	48 85 118 150 179	22.310	1.84 E-012
Living Room	2	87, 145	15.999	1.15 E-12
	3	76, 123, 163	18.197	6.92 E-12
	4	56, 97, 132, 168	20.673	9.22 E-12
	5	49, 88, 120, 146, 178	22.225	2.86 E-02
Blonde	2	106, 155	14.609	6.92 E-13
	3	53, 112, 158	19.157	9.23 E-13
	4	50, 103, 139, 168	20.964	5.48 E-01
	5	49, 92, 121, 152, 172	22.409	6.50 E-01
Bridge	2	91, 56	13.943	4.61 E-13
	3	72, 120, 177	17.019	7.10 E-01
	4	63, 103, 145, 193	18.872	2.91 E-01
	5	56, 91, 124, 159, 201	20.539	3.57 E-01
Butterfly	2	99, 151	13.934	7.30 E-02
	3	82, 119, 160	16.932	6.17 E-01
	4	71, 102, 130, 163	19.259	3.07 E+00
	5	62, 77, 109, 137, 167	21.450	3.87 E+00
Lake	2	85, 154	14.638	4.61 E-13
	3	78, 140, 194	15.860	1.84 E-12
	4	67, 110, 158, 198	17.629	2.68 E-01
	5	57 88 127 166 200	19.416	1.12 E-01

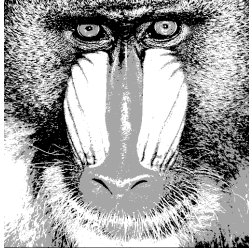
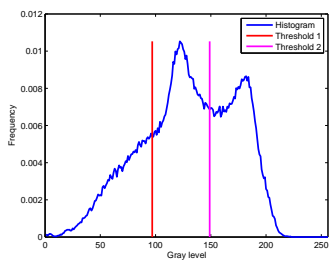
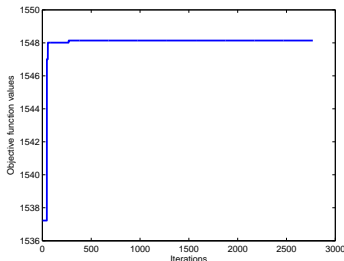
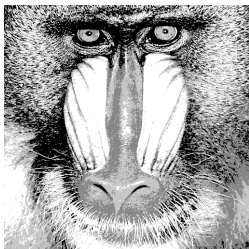
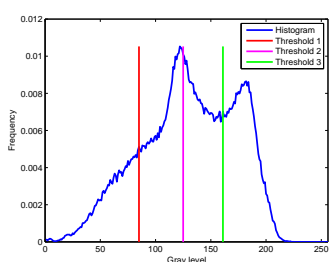
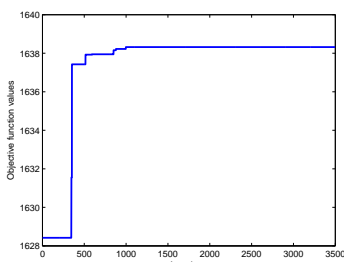
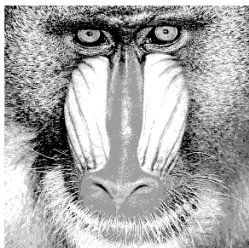
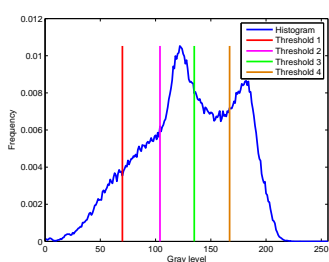
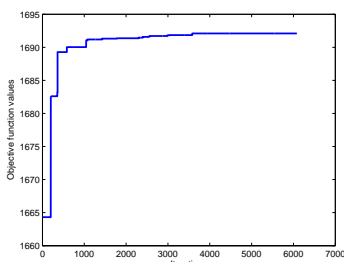
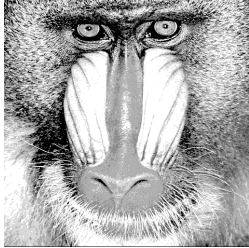
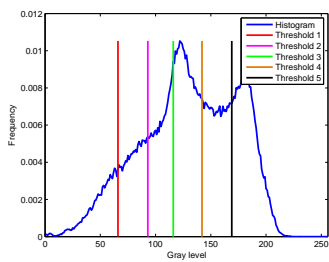
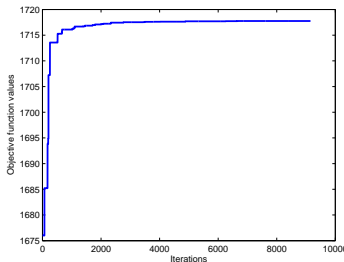
**Tabla 4.3:** Resultados de aplicar el algoritmo HSA usando la función de Otsu al conjunto de imágenes de prueba.



**Tabla 4.4:** Resultados de aplicar el algoritmo HSA usando la función de Otsu a la imagen de prueba: *Camerman*.

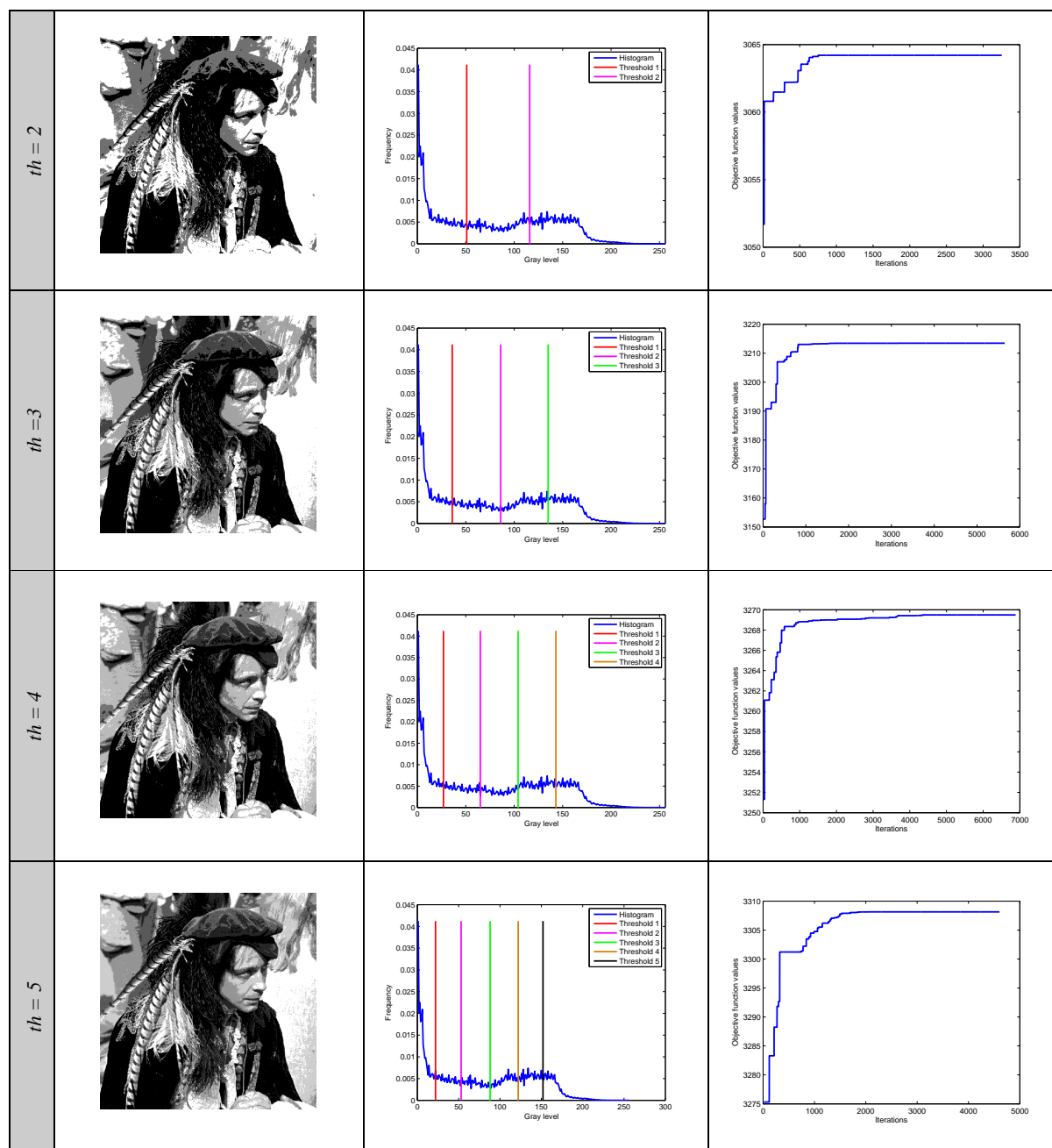
$th = 2$			
$th = 3$			
$th = 4$			
$th = 5$			

**Tabla 4.5:** Resultados de aplicar el algoritmo HSA usando la función de Otsu a la imagen de prueba: Lena.

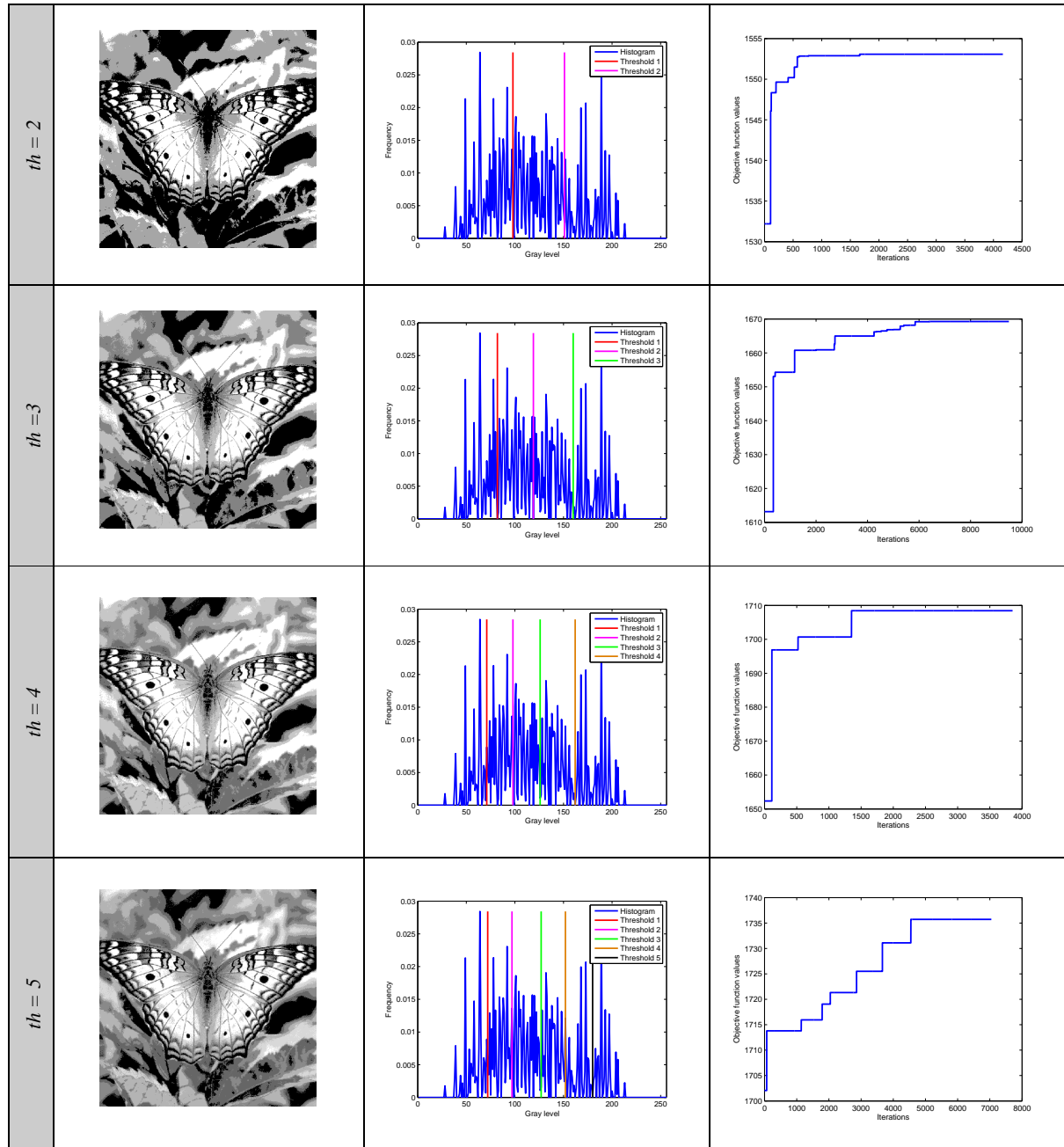
$th = 2$			
$th = 3$			
$th = 4$			
$th = 5$			

**Tabla 4.6:** Resultados de aplicar el algoritmo HSA usando la función de Otsu a la imagen de prueba: *Baboon*.





**Tabla 4.7:** Resultados de aplicar el algoritmo HSA usando la función de Otsu a la imagen de prueba: *Hunter*.



**Tabla 4.8:** Resultados de aplicar el algoritmo HSA usando la función de Otsu a la imagen de prueba: *Butterfly*.

#### 4.3.4.2 Resultados de HSA usando la ecuación de Kapur

En esta sección se analiza el rendimiento del algoritmo HSA considerando como función objetivo la entropía propuesta en la ecuación (4.21), la cual corresponde al método de Kapur [65]. En la Tabla 4.9, se presentan los resultados experimentales obtenidos después de aplicar el algoritmo de segmentación HSA y la ecuación de Kapur al conjunto de imágenes de prueba. Los valores listados son PSNR, STD y los mejores valores de umbral de la última población de HSA ( $\mathbf{x}_{best}^c$ ). Los experimentos han sido desarrollados usando los mismos criterios que en el método de Otsu, es decir, el algoritmo HSA ha sido configurado con los parámetros usados en la sección 4.3.4.1; del mismo modo, el criterio de parada será un porcentaje del valor total de iteraciones.

Los resultados de aplicar los mejores umbrales a las imágenes de prueba seleccionadas, se presentan en las Tablas 4.10, 4.11, 4.12, 4.13 y 4.14. De manera similar al método de Otsu, se usan cuatro diferentes puntos de umbral  $th = 2, 3, 4, 5$ . En todas las tablas se muestra la imagen segmentada, el histograma con los respectivos umbrales y la evolución del valor de la función objetivo a lo largo de las iteraciones del algoritmo HSA.

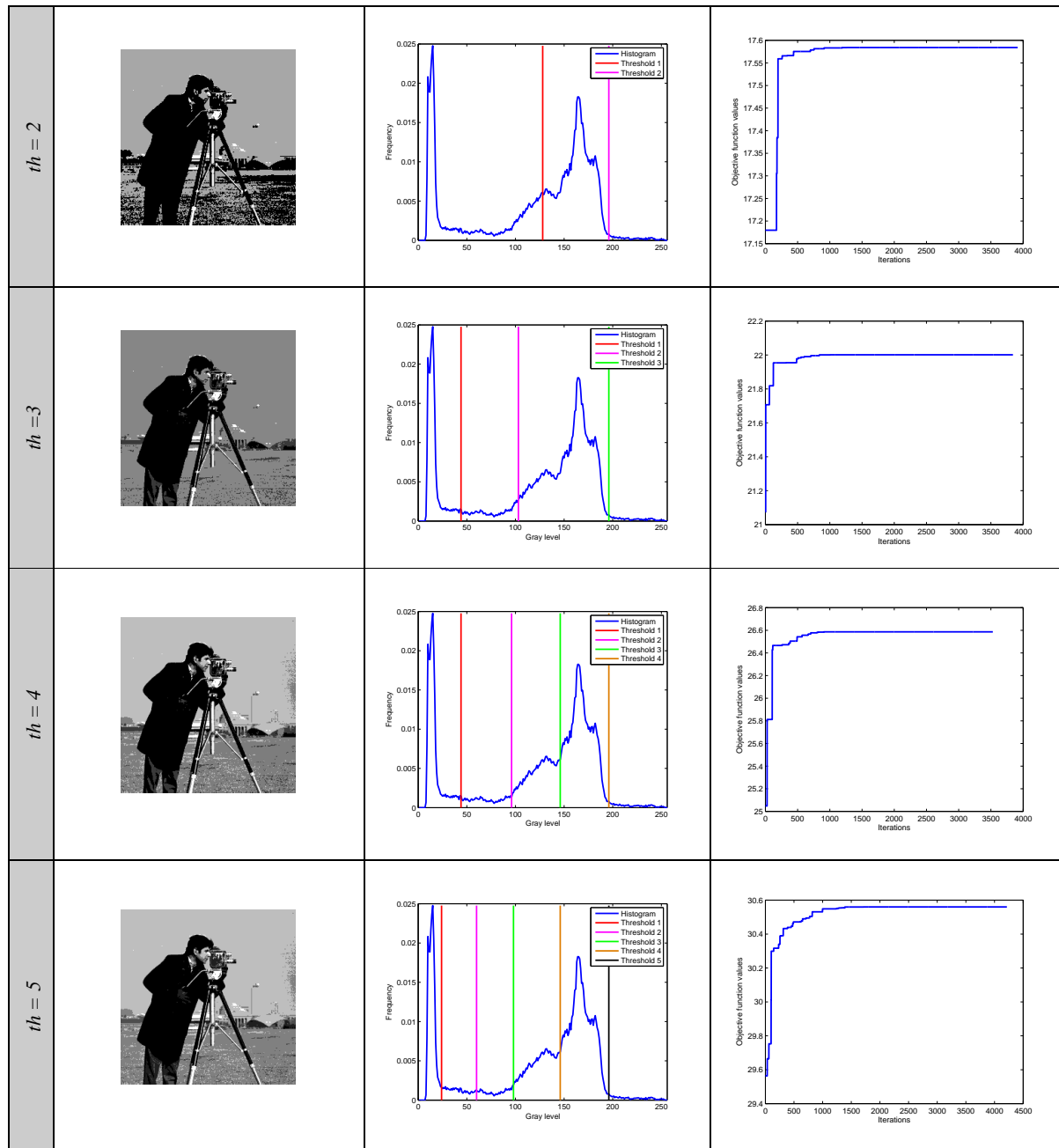
A partir de los resultados de los métodos de Otsu y Kapur, es posible observar que el algoritmo HSA converge o se estabiliza después de un cierto número de iteraciones, este valor depende de la cantidad de umbrales a encontrar. Con fines experimentales, se deja correr el algoritmo hasta que el criterio de parada se cumple. De este modo, las gráficas muestran claramente que la convergencia se alcanza en las primeras iteraciones del proceso de optimización. Por otra parte, las imágenes segmentadas evidencian que es mejor ser segmentadas con cuatro o cinco valores de umbral, sin embargo si la aplicación no requiere ser extremadamente exacta, es posible seleccionar  $th = 3$ .

#### 4.3.4.3 Comparaciones


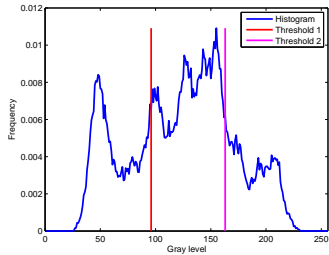
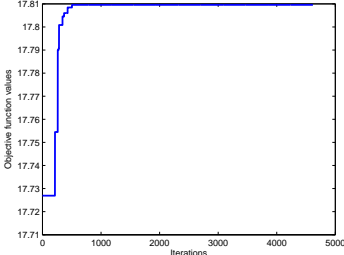

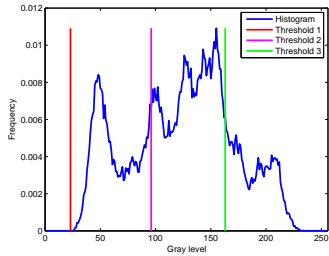
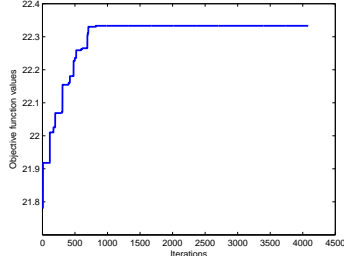

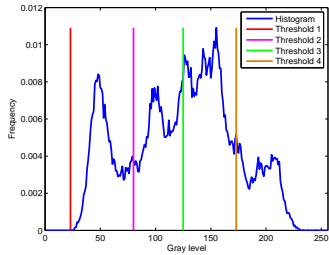
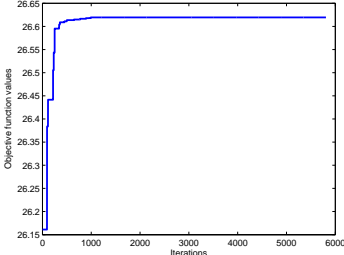

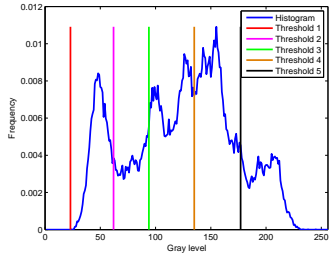
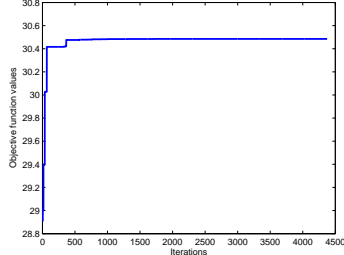
Para analizar los resultados del algoritmo de segmentación basado en HSA se han realizado dos diferentes comparaciones. La primera involucra la comparación entre las dos versiones del enfoque propuesto, una con la función de Otsu y otra con la función de Kapur. Se analizan los resultados de la comparación entre HSA y otros enfoques propuestos en la literatura.

Imagen	$k$	Umbrales $\mathbf{x}_{best}^c$	PSNR	STD
Cameraman	2	128, 196	13.626	3.60 E-15
	3	44, 103, 196	14.460	1.40 E-03
	4	44, 96, 146, 196	20.153	1.20 E-03
	5	24, 60, 98, 146, 196	20.661	2.75 E-02
Lena	2	96, 163	14.638	3.60 E-15
	3	23, 96, 163	16.218	7.66 E-02
	4	23, 80, 125, 173	19.287	1.44 E-14
	5	23, 71, 109, 144, 180	21.047	1.22 E-02
Baboon	2	79, 143	16.016	1.08 E-14
	3	79, 143, 231	16.016	7.19 E-02
	4	44, 98, 152, 231	18.485	8.47 E-02
	5	33, 74, 114, 159, 231	20.507	1.08 E-14
Hunter	2	92, 179	15.206	1.44 E-14
	3	59, 117, 179	18.500	4.16 E-04
	4	44, 89, 133, 179	21.065	4.31 E-04
	5	44, 89, 133, 179, 222	21.086	3.43 E-02
Airplane	2	70, 171	15.758	3.30 E-03
	3	68, 126, 182	18.810	1.08 E-14
	4	68, 126, 182, 232	18.810	1.82 E-01
	5	64, 104, 143, 184, 232	20.321	1.80 E-01
Peppers	2	66, 143	16.265	7.21 E-15
	3	62, 112, 162	18.367	1.80 E-14
	4	62, 112, 162, 227	18.376	2.39 E-02
	5	48, 86, 127, 171, 227	18.827	4.17 E-04
Living Room	2	89, 170	14.631	1.40 E-03
	3	47, 103, 175	17.146	2.70 E-03
	4	47, 98, 149, 197	19.144	1.34 E-02
	5	42, 85, 125, 162, 197	21.160	1.89 E-02
Blonde	2	125, 203	12.244	1.44 E-14
	3	65, 134, 203	16.878	1.00 E-03
	4	65, 113, 155, 203	20.107	5.50 E-03
	5	65, 113, 155, 203, 229	20.107	4.48 E-02
Bridge	2	94, 171	13.529	7.40 E-03
	3	65, 131, 195	16.806	1.44 E-02
	4	53, 102, 151, 199	18.902	2.47 E-02
	5	40, 85, 131, 171, 211	20.268	2.12 E-02
Butterfly	2	27, 213	8.1930	2.25 E-02
	3	27, 120, 213	13.415	8.60 E-04
	4	27, 96, 144, 213	16.725	3.80 E-03
	5	27, 83, 118, 152, 213	19.413	3.90 E-03
Lake	2	91, 163	14.713	1.44 E-14
	3	73, 120, 170	16.441	3.05 E-04
	4	69, 112, 156, 195	17.455	4.53 E-02
	5	62, 96, 131, 166, 198	18.774	3.66 E-02

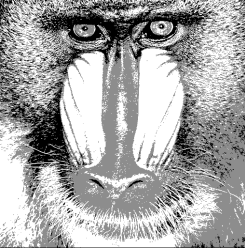
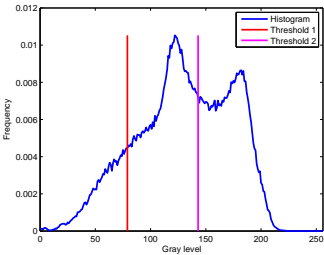
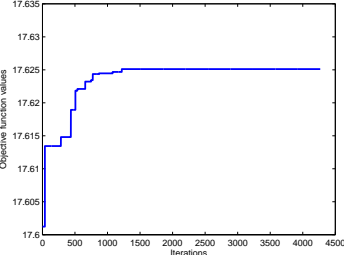
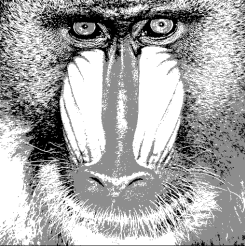
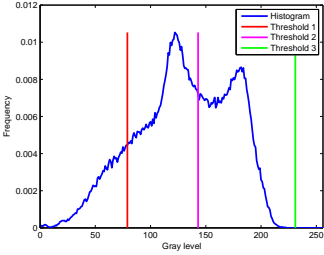
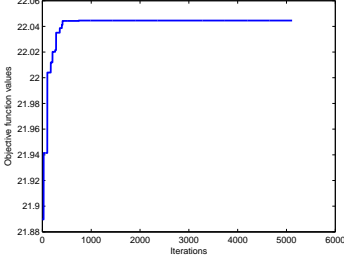
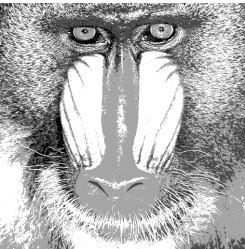
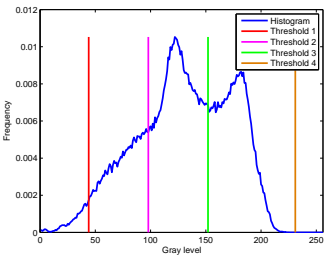
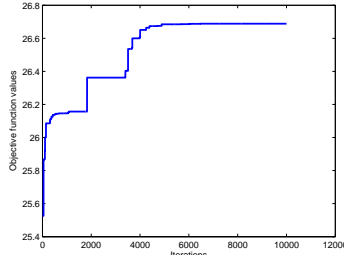
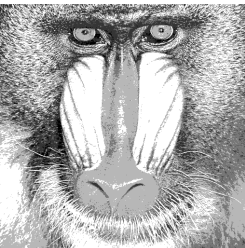
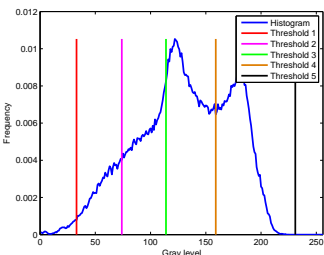
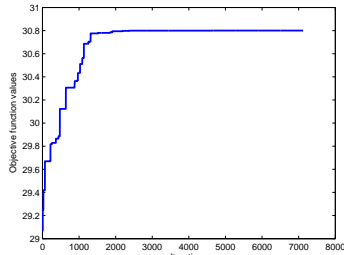
**Tabla 4.9:** Resultados de aplicar el algoritmo HSA usando la función de Kapur al conjunto de imágenes de prueba.



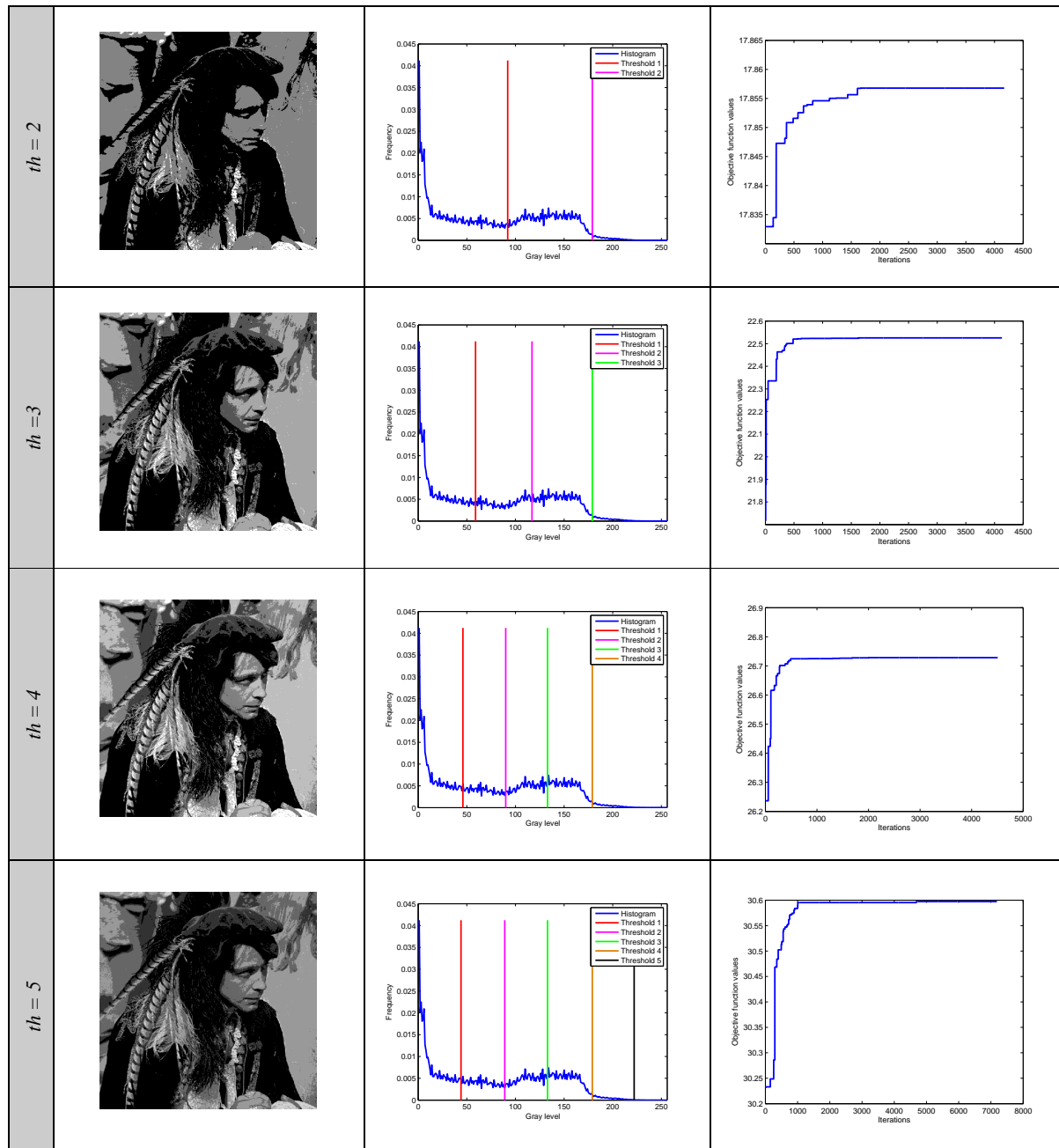
**Tabla 4.10:** Resultados de aplicar el algoritmo HSA usando la función de Kapur a la imagen de prueba: *Cameraman*.

$th = 2$			
$th = 3$			
$th = 4$			
$th = 5$			

**Tabla 4.11:** Resultados de aplicar el algoritmo HSA usando la función de Kapur a la imagen de prueba: Lena.

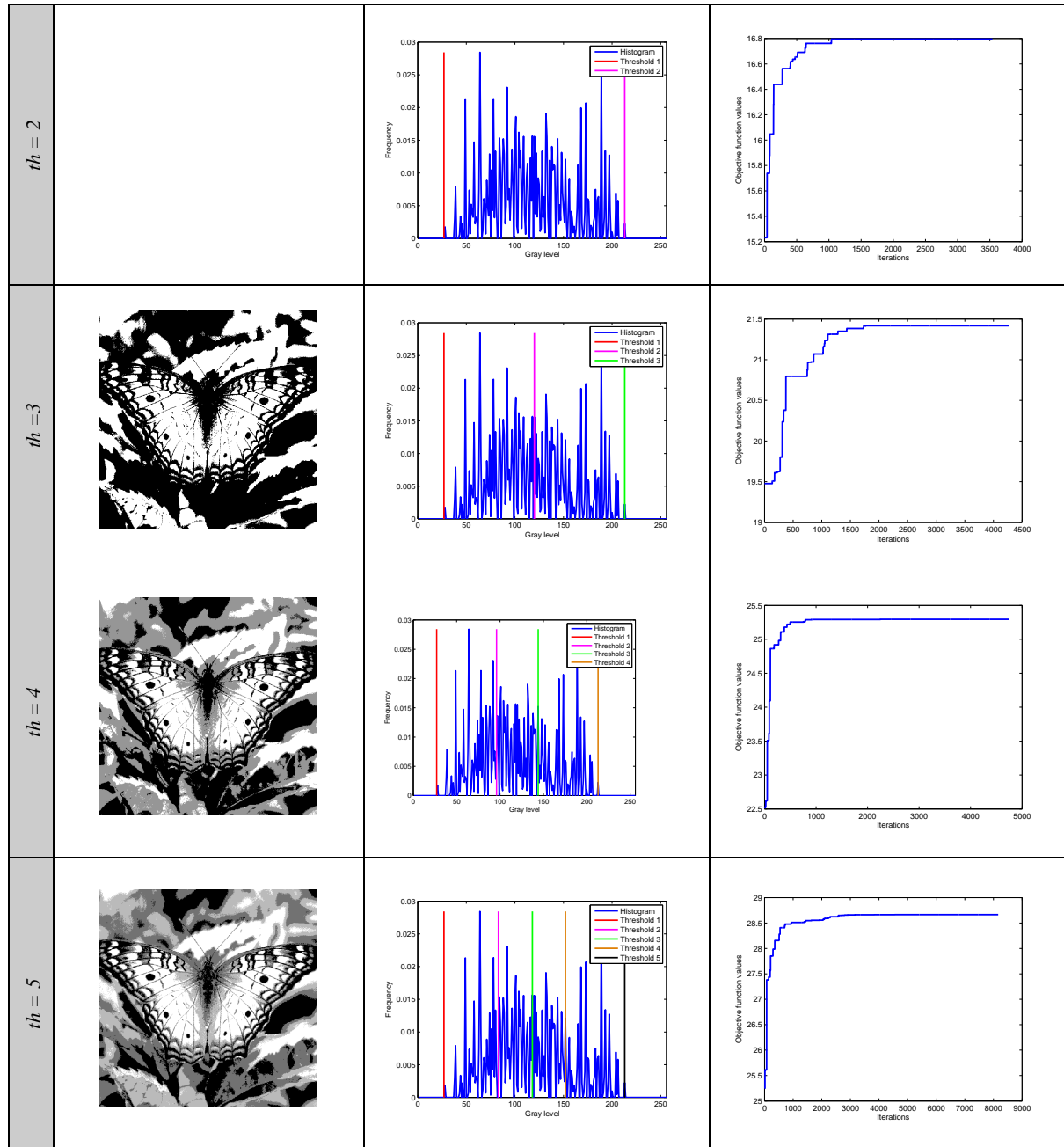
$th = 2$			
$th = 3$			
$th = 4$			
$th = 5$			

**Tabla 4.12:** Resultados de aplicar el algoritmo HSA usando la función de Kapur a la imagen de prueba: *Baboon*.



**Tabla 4.13:** Resultados de aplicar el algoritmo HSA usando la función de Kapur a la imagen de prueba: *Hunter*.





**Tabla 4.14:** Resultados de aplicar el algoritmo HSA usando la función de Kapur a la imagen de prueba: *Butterfly*.

*Comparaciones entre HSA usando Otsu y Kapur:* con la finalidad de comparar estadísticamente los resultados de las Tablas 4.3 y 4.9, se ha realizado una prueba no paramétrica conocida como la prueba de rangos de Wilcoxon [174, 175], para la cual se emplean 35 experimentos distintos en cada imagen y para cada número de puntos umbral. El hecho de realizar esta prueba permite evaluar las diferencias en los resultados de los dos métodos. El análisis se lleva a cabo considerando un nivel de significativo del 5% sobre el valor de PSNR. En la Tabla 4.15 se reportan los valores  $P$  obtenidos por la prueba de Wilcoxon, donde se hizo una comparación por pares entre los valores de PSNR usando las funciones de Otsu y Kapur en conjunto con el algoritmo HSA. Como una hipótesis nula, se asume que no existe diferencia entre los valores de ambas funciones, mientras que una hipótesis alternativa, considera que los resultados de ambos métodos son distintos. Todos los valores  $P$  reportados en la Tabla 4.15 son menores que 0.05 o lo que es lo mismo el 5% del nivel de significancia, lo cual es una fuerte evidencia de que la hipótesis nula no se cumple e indica que los valores de PSNR obtenidos usando Otsu y HSA son estadísticamente mejores y no son obtenidos por casualidad.

*Comparaciones ente MTHSA y otras técnicas de umbralado:* los resultados obtenidos por el algoritmo MTHSA se han comparado con los obtenidos por otros métodos de umbralado encontrados en la literatura, dichas técnicas están basadas en: algoritmos genéticos (GA) [191], optimización por enjambre de partículas (PSO) [68] y algoritmo de búsqueda de alimento bacteriano (BFA) [70].

Todos los algoritmos se ejecutan 35 veces en cada una de las imágenes seleccionadas. Dichas imágenes son las mismas que se han usado para mostrar gráficamente los resultados de la umbralización (*Cameraman*, *Lena*, *Baboon*, *Hunter and Butterfly*). Para cada imagen se calcula el PSNR, STD y el promedio del valor de la función objetivo. Además, todo el conjunto de imágenes se prueba con las funciones de Otsu y Kapur. En la Tabla 4.16 se presentan los valores calculados para cada una de las cinco imágenes de prueba. Aquí queda claro que el algoritmo basado en HSA provee un mejor desempeño en comparación con sus contrapartes. Los resultados que se analizan y presentan en esta tabla, corresponden a la implementación de HSA empleando como función objetivo la ecuación de Otsu.

Por otra parte, el mismo experimento se llevó a cabo usando el método de Kapur. Usando el mismo criterio que con Otsu, el algoritmo se ejecuta 35 veces por cada imagen,

finalmente los resultados se muestran en la Tabla 4.17, donde es posible apreciar que el método propuesto MTHSA es mejor en comparación con GA, PSO y BFA.

Imagen	$k$	Valores $P$ Otsu vs. Kapur
<i>Cameraman</i>	2	1.0425 E-16
	3	2.1435 E-15
	4	2.6067 E-16
	5	6.2260 E-16
Lena	2	1.0425 E-16
	3	9.4577 E-15
	4	9.7127 E-15
	5	1.2356 E-12
<i>Baboon</i>	2	1.0425 E-16
	3	1.7500 E-02
	4	5.3417 E-14
	5	1.4013 E-14
<i>Hunter</i>	2	1.0425 E-16
	3	2.6067 E-16
	4	6.6386 E-14
	5	6.4677 E-15
<i>Airplane</i>	2	1.0425 E-16
	3	4.6500 E-02
	4	1.7438 E-13
	5	6.2475 E-13
<i>Peppers</i>	2	1.0425 E-16
	3	1.0425 E-16
	4	6.0498 E-15
	5	5.3194 E-15
<i>Living Room</i>	2	3.2405 E-15
	3	1.4223 E-15
	4	1.3175 E-14
	5	6.2220 E-14
<i>Blonde</i>	2	1.0425 E-16
	3	2.6067 E-16
	4	4.0480 E-13
	5	7.8167 E-04
<i>Bridge</i>	2	7.3588 E-06
	3	1.1300 E-04
	4	1.9400 E-02
	5	2.1900 E-02
<i>Butterfly</i>	2	1.1615 E-14
	3	2.5697 E-14
	4	3.7190 E-13
	5	1.7941 E-06
<i>Lake</i>	2	1.0425 E-16
	3	5.6120 E-16
	4	9.8174 E-14
	5	2.2292 E-14

**Tabla 4.15:** Valores  $P$  obtenidos por la prueba de Wilcoxon, comparando los resultados de PSNR de Otsu contra los de Kapur.

Imagen	$k$	MTHSA			GA			PSO			BF		
		PSNR	STD	Media	PSNR	STD	Media	PSNR	STD	Media	PSNR	STD	Media
Camera man	2	17.247	2.30 E-12	3651.9	17.048	0.0232	3604.5	17.033	0.0341	3598.3	17.058	0.0345	3590.9
	3	20.211	1.55 E-02	3727.4	17.573	0.1455	3678.3	19.219	0.2345	3662.7	20.035	0.2459	3657.5
	4	21.533	2.76 E-12	3782.4	20.523	0.2232	3781.5	21.254	0.3142	3777.4	21.209	0.4560	3761.4
	5	23.282	5.30 E-03	3813.7	21.369	0.4589	3766.4	22.095	0.5089	3741.6	22.237	0.5089	3789.8
Lena	2	15.401	9.22 E-13	1964.4	15.040	0.0049	1960.9	15.077	0.0033	1961.4	15.031	2.99 E-04	1961.5
	3	17.427	2.99 E-02	2131.4	17.304	0.1100	2126.4	17.276	0.0390	2127.7	17.401	0.0061	2128.0
	4	18.763	2.77 E-01	2194.9	17.920	0.2594	2173.7	18.305	0.1810	2180.6	18.507	0.0081	2189.0
	5	19.443	3.04 E-01	2218.7	18.402	0.3048	2196.2	18.770	0.2181	2212.5	19.001	0.0502	2215.6
Baboon	2	15.422	6.92 E-13	1548.1	15.304	0.0031	1547.6	15.088	0.0077	1547.9	15.353	8.88 E-04	1548.0
	3	17.709	1.92 E-02	1638.3	17.505	0.1750	1633.5	17.603	0.0816	1635.3	17.074	0.0287	1637.0
	4	20.289	5.82 E-02	1692.1	18.708	0.2707	1677.7	19.233	0.0853	1684.3	19.654	0.0336	1690.7
	5	21.713	4.40 E-01	1717.5	20.203	0.3048	1712.9	20.526	0.1899	1712.9	21.160	0.1065	1716.7
Hunter	2	17.875	2.30 E-12	3054.2	17.088	0.0470	3064.1	17.932	0.2534	3064.1	17.508	0.0322	3064.1
	3	20.350	2.30 E-12	3213.4	20.045	0.1930	3212.9	19.940	0.9727	3212.4	20.350	0.9627	3213.4
	4	22.203	1.22 E-02	3269.5	20.836	0.6478	3268.4	21.128	2.2936	3266.3	21.089	2.2936	3266.3
	5	23.703	1.84 E-12	3308.1	21.284	1.6202	3305.6	22.026	4.1811	3276.3	22.804	3.6102	3291.1
Butterfly	2	13.934	7.30 E-02	1553.0	13.007	0.0426	1553.0	13.092	0.0846	1553.0	13.890	0.0643	1553.0
	3	16.932	6.17 E-01	1669.2	15.811	0.3586	1669.0	17.261	2.6268	1665.7	17.285	1.2113	1667.2
	4	19.259	3.07 E+00	1708.3	17.104	0.6253	1709.9	17.005	3.7976	1702.9	17.128	2.2120	1707.0
	5	21.450	3.87 E+00	1728.0	18.593	0.5968	1734.4	18.099	6.0747	1730.7	18.9061	3.5217	1733.0

**Tabla 4.16:** Comparaciones entre los algoritmos MTHSA, GA, PSO y BFA aplicados sobre las imágenes de prueba seleccionadas usando el método de Otsu.

Imagen	$k$	HSAMA			GA			PSO			BF		
		PSNR	STD	Media	PSNR	STD	Media	PSNR	STD	Media	PSNR	STD	Media
Camera man	2	13.626	3.60 E-15	17.584	11.941	0.1270	15.341	12.259	0.1001	16.071	12.264	0.0041	16.768
	3	14.460	1.40 E-03	22.007	14.827	0.2136	20.600	15.211	0.1107	21.125	15.250	0.0075	21.498
	4	20.153	1.20 E-03	26.586	17.166	0.2857	24.267	18.000	0.2005	25.050	18.406	0.0081	25.093
	5	20.661	2.75 E-02	30.553	19.795	0.3528	28.326	20.963	0.2734	28.365	21.211	0.0741	30.026
Lena	2	14.638	3.60 E-15	17.809	12.334	0.0049	16.122	12.345	0.0033	16.916	12.345	2.99 E-4	16.605
	3	16.218	7.66 E-02	22.306	14.995	0.1100	20.920	15.133	0.0390	20.468	15.133	0.0061	20.812
	4	19.287	1.44 E-14	26.619	17.089	0.2594	23.569	17.838	0.1810	24.449	17.089	0.0081	26.214
	5	21.047	1.22 E-02	30.485	19.549	0.3043	27.213	20.442	0.2181	27.526	19.549	0.0502	28.046
Baboon	2	16.016	1.08 E-14	17.625	12.184	0.0567	16.425	12.213	0.0077	16.811	12.216	8.88 E-4	16.889
	3	16.016	7.19 E-02	22.117	14.745	0.1580	21.069	15.008	0.0816	21.088	15.211	0.0287	21.630
	4	18.485	8.47 E-02	26.671	16.935	0.1765	25.489	17.574	0.0853	24.375	17.999	0.0336	25.446
	5	20.507	1.08 E-14	30.800	19.662	0.2775	29.601	20.224	0.1899	30.994	20.720	0.1065	30.887
Hunter	2	15.206	1.44 E-14	17.856	12.349	0.0148	16.150	12.370	0.0068	15.580	12.373	0.0033	16.795
	3	18.500	4.16 E-04	22.525	14.838	0.1741	21.026	15.128	0.0936	20.639	15.553	0.1155	21.860
	4	21.065	4.31 E-04	26.728	17.218	0.2192	25.509	18.040	0.1560	27.085	18.381	0.0055	26.230
	5	21.086	3.43 E-02	30.612	19.563	0.3466	29.042	20.533	0.2720	29.013	21.256	0.0028	28.856
Butterfly	2	8.1930	2.25 E-02	16.791	10.470	0.0872	15.481	10.474	0.0025	14.098	10.474	0.0014	15.784
	3	13.415	8.60 E-04	21.417	11.628	0.2021	20.042	12.313	0.1880	19.340	12.754	0.0118	21.308
	4	16.725	3.80 E-03	25.292	13.314	0.2596	23.980	14.231	0.2473	25.190	14.877	0.0166	25.963
	5	19.413	3.90 E-03	28.664	15.756	0.3977	27.411	16.337	0.2821	27.004	16.828	0.0877	27.980

**Tabla 4.17:** Comparaciones entre los algoritmos MTHSA, GA, PSO y BFA aplicados sobre las imágenes de prueba seleccionadas usando el método de Kapur.

## 4.4 Resumen

En este capítulo se introduce el algoritmo de optimización *Harmony Search* (HSA), el cual es una técnica metaheurística inspirada en la improvisación musical. Los operadores modifican las posibles soluciones en búsqueda de la mejor armonía, para esto se emplea la función objetivo, cuyos valores guían a la población hacia el óptimo global. Además de describir los pasos del HSA también se presenta una implementación que da solución a un problema de procesamiento de imágenes conocido como umbralización multinivel. De este modo, el algoritmo HSA busca los mejores valores de umbral que maximicen la función objetivo, la cual depende de si se elige el método de Otsu o el de Kapur. El algoritmo propuesto basado en HSA ha sido probado sobre un conjunto de imágenes que comúnmente son usadas para el problema de umbralización. Los resultados obtenidos para cada imagen son evaluados usando el PSNR, que en este caso mide la calidad que tiene la imagen segmentada en comparación con la original. Como la mayoría de los algoritmos metaheurísticos, HSA requiere que algunos parámetros sean calibrados para obtener mejores resultados, aquí se propone una calibración óptima en base a diferentes experimentos realizados sobre una de las imágenes del conjunto de prueba.

Para comprobar la eficiencia del HSA para segmentación de imágenes, se han realizado dos diferentes comparaciones, la primera de ellas es una prueba no paramétrica que comprueba que el método de Otsu obtiene mejores resultados de PSNR. La segunda comparación demuestra que HSA consigue mejores resultados de PSNR que otros algoritmos similares basados en técnicas metaheurísticas.

# Capítulo 5

## Algoritmo de optimización *Artificial Bee Colony:* diseño de celdas solares

---

En este capítulo se realiza un estudio sobre el algoritmo de optimización *Artificial Bee Colony* (ABC), en él se describen cada una de sus etapas, los operadores que emplea y la configuración de sus parámetros. Esta técnica se implementa para el diseño de celdas solares y con esto se verifica su efectividad para resolver problemas de optimización en ingeniería. Primero se detalla la teoría referente al algoritmo ABC, aquí se presenta el perfil biológico, los operadores usados para la optimización y la configuración de sus parámetros. A continuación, se describen los modelos de celdas solares, más concretamente las funciones objetivo que serán usadas para cada uno de ellos. Seguidamente se realiza el diseño para la implementación del ABC con el fin de encontrar los valores óptimos que modelen las celdas solares. Por último, se presentan los resultados experimentales que respaldan la implementación de ABC y sus capacidades para el diseño de celdas solares.

## 5.1 Algoritmo de optimización *Artificial Bee Colony* (ABC)

El modelo mínimo de una colonia de abejas consta de tres clases de abejas: obreras, espectadoras y exploradoras. Las abejas obreras son las responsables de investigar dónde existen fuentes de alimento y además deben compartir la información con las abejas espectadoras. La fuente de alimento que posea la mejor calidad, tendrá mayor posibilidad de ser seleccionada por las abejas espectadoras que la que tiene una baja calidad. Una abeja obrera, que ha explorado fuentes de alimento de baja calidad es rechazada por las abejas obreras y las espectadoras, por tal motivo pasará a ser una abeja exploradora y buscará aleatoriamente nuevas fuentes de alimento. Por lo tanto, la explotación de las fuentes de alimento es conducida por las abejas obreras y las espectadoras, mientras que las abejas exploradoras se dedican solamente a la exploración.

### 5.1.1 Descripción del algoritmo ABC

De forma similar a otros enfoques basados en enjambre, el algoritmo ABC se fundamenta en un proceso iterativo, que se inicia con una población de soluciones generadas aleatoriamente las cuales se denominan fuentes de alimento. Para llevar a cabo el proceso de optimización, se aplican tres operaciones hasta que se alcanza el criterio de parada [38]:

1. Enviar las abejas obreras.
2. Seleccionar las mejores fuentes de alimento usando las abejas espectadoras.
3. Determinar o seleccionar las abejas exploradoras.

### 5.1.2 Inicialización de la población

El primer paso del algoritmo consiste en inicializar la población de  $N_p$  fuentes de alimento. Cada una de ellas es un vector de  $d$  dimensiones, que contiene los valores que serán optimizados. Cada vector es inicializado aleatoriamente tomando valores uniformemente distribuidos en un espacio de búsqueda. La ecuación (5.1) describe el proceso de inicialización.

$$x_{i,j} = l_j + \text{rand}(0,1) \cdot (u_j - l_j), \quad (5.1)$$

$$j = 1, 2, \dots, d; \quad i = 1, 2, \dots, N_p$$

donde  $x_{i,j}$  es una fuente de alimento, el índice  $i$  corresponde a la  $i$ -ésima fuente de alimento,  $\text{rand}(0,1)$  corresponde a la función que genera valores aleatorios uniformemente distribuidos entre  $[0,1]$ . Finalmente  $l_j$  y  $u_j$  son los límites inferior y superior de cada dimensión.

### 5.1.3 Enviar abejas obreras

Las abejas obreras se usan para generar nuevas soluciones, la cantidad de abejas de este tipo es igual al número de fuentes de alimento. De acuerdo con la literatura [36, 37] la población entera se divide en dos ( $N_p/2$ ), una parte corresponde a las abejas obreras y el resto a las abejas espectadoras. Esta división opera como parte de la estrategia de búsqueda. Por tal motivo, es necesario que se aplique a todos los problemas donde ABC se emplea como estrategia de optimización. A continuación se muestra cómo son enviadas las abejas obreras:

$$\begin{aligned} B_{i,j} &= x_{i,j} + \phi_{i,j} (x_{i,j} - x_{k,j}), \quad \forall i \neq k \\ k &\in \text{rand}\{1, N_p\}, \quad j \in \{1, 2, \dots, d\} \end{aligned} \quad (5.2)$$

donde el parámetro  $\phi_{j,i}$  es un valor aleatorio seleccionado en el rango de  $[-1, 1]$ , como ya se mencionó,  $i$  corresponde a la  $i$ -ésima fuente de alimento y  $j$  es el índice de la dimensión del problema. Entonces para generar una nueva fuente de alimento usando el operador de abejas obreras, se selecciona aleatoriamente una fuente de alimento  $k$  en la dimensión  $j$ . Si algún parámetro de la fuente de alimento encontrada por la abeja obrera  $B_{i,j}$  excede los límites, ésta será ajustada para que se sitúe dentro del rango apropiado. Después de este proceso se calcula el valor de afinidad asociado a cada solución, el cual se usa para evaluar la calidad de una fuente de alimento. Para un problema de minimización el valor de afinidad se obtiene usando la siguiente ecuación:

$$\text{fit}_i = \begin{cases} \frac{1}{1 + J_i} & \text{if } J_i \geq 0 \\ 1 + \text{abs}(J_i) & \text{if } J_i < 0 \end{cases} \quad (5.3)$$

donde  $J_i$  es el valor de la función objetivo para la solución candidata  $\mathbf{x}_i$ . Dentro del contexto de esta investigación,  $J_i$  representa el error cuadrático medio (RMSE) que será definido en la siguiente sección. El siguiente proceso consiste en aplicar una selección



voraz entre los valores de las abejas obreras contenidas en  $\mathbf{B}_i$  y el vector que contiene las fuentes de alimento iniciales  $\mathbf{x}_i$ . Esto significa que si la cantidad de néctar (valor de afinidad) de  $\mathbf{B}_i$  es mejor, entonces la solución  $\mathbf{x}_i$  es remplazada por  $\mathbf{B}_i$ , en caso contrario el valor de  $\mathbf{x}_i$  prevalece.

#### 5.1.4 Selección de las fuentes de alimento por medio de las abejas espectadoras

Para describir la fase de las abejas espectadoras, primero es necesario explicar que el número de estas abejas corresponde a la cantidad de fuentes de alimento existentes. De esta forma, las fuentes de alimento son modificadas varias veces dependiendo del valor de afinidad obtenido mediante la ecuación (5.3). Para que una fuente de alimento pueda ser elegida, primero es necesario calcular un factor de probabilidad, el cual es obtenido en base a la afinidad, como se muestra a continuación:

$$\text{Prob}_i = \frac{fit_i}{\sum_{i=1}^{N_p} fit_i} \quad (5.4)$$

donde,  $fit_i$  corresponde al valor de afinidad de la  $i$ -ésima fuente de alimento y tiene relación con el valor objetivo de dicho elemento de la población. Si el valor de la función objetivo se incrementa, entonces la probabilidad de que una fuente de alimento sea seleccionada por una abeja espectadora es mayor. Cuando una fuente de alimento se selecciona, se obtiene un nuevo valor usando la ecuación (5.2), el valor de afinidad se calcula y se aplica el proceso de selección voraz para modificar (o no) su posición.

#### 5.1.5 Determinar las abejas exploradoras

El paso final, corresponde al de las abejas exploradoras, las cuales sólo se aplican si la fuente de alimento  $i$  no puede ser mejorada en un determinado número de intentos. Entonces, la fuente de alimento se considera como abandonada y en lugar de ser modificada por una abeja espectadora se modifica con una exploradora usando la ecuación (5.1). Para verificar el número de intentos en que será modificada la fuente de alimento, se establece un “límite” que es un contador asignado a cada elemento de la población y se incrementa cada vez que el valor de afinidad no se modifica.

### 5.1.6 Algoritmo ABC

El algoritmo de optimización basado en colonias de abejas se puede resumir en los pasos mostrados en el Algoritmo 5.1.

---

**Algoritmo 5.1: ABC**


---

```

1: Generar la población inicial  $\mathbf{X}$ 
2: Calcular  $f(\mathbf{X})$ 
3:  $ciclo \leftarrow 1$ 
4: while  $ciclo \leq max\_ciclo$  do
5:   for  $i = 1$  to  $N_p/2$  do
6:     Producir una nueva solución  $B_i$  usando la ecuación (5.2)
7:     Calcular  $fit_i$ 
8:     Aplicar el proceso de selección voraz
9:   endfor
10:  Calcular  $Prob_i$ 
11:  for  $i = 1$  to  $N_p/2$  do
12:    Seleccionar una nueva solución dependiendo de  $Prob_i$ 
13:    Producir una nueva solución  $B_i$ 
14:    Calcular  $fit_i$ 
15:    Aplicar el proceso de selección voraz
16:  endfor
17:  if (una solución es abandonada) then
18:    Generar una nueva solución usando la ecuación (5.1)
19:  endif
20:  Almacenar la mejor solución actual
21:   $ciclo \leftarrow ciclo + 1$ 
22: endwhile

```

---

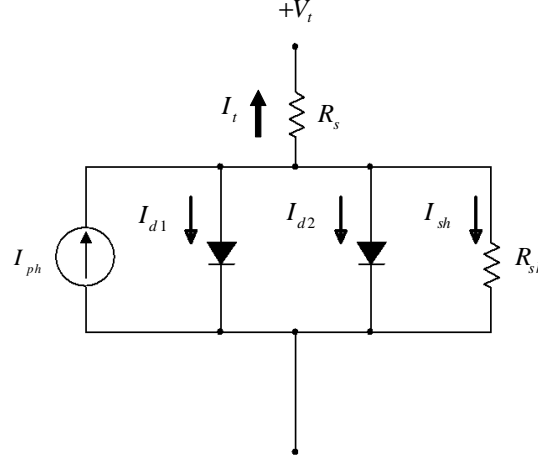
## 5.2 Diseño de celdas solares

El modelado de celdas solares o fotovoltaicas consta de dos pasos: la formulación del modelo matemático y la precisa estimación de sus parámetros. Como se mencionó en el capítulo dos, en la literatura existen básicamente dos modelos: diodo simple (DS) y diodo doble (DD) [89]. A continuación se describen estos modelos, a partir de los cuales se desprenden dos funciones objetivo.

### 5.2.1 Modelo de doble diodo

Las celdas solares son idealmente modeladas considerando una fuente de corriente foto-generada ( $I_{ph}$ ), la cual es derivada por un diodo rectificador. Sin embargo, en términos prácticos la fuente de corriente  $I_{ph}$  es derivada por otro diodo el cual modela la corriente de recombinación de carga espacial y otras condiciones no ideales. El modelo de una celda

solar también incluye un resistor conectado en serie con los elementos de derivación de la celda [89]. La Figura 5.1 muestra el circuito equivalente para el modelo de doble diodo.



**Figura 5.1:** Modelo de doble diodo para las celdas solares.

De acuerdo con la Figura 5.1, la corriente en el terminal de medición de la celda solar se puede calcular como se muestra a continuación:

$$I_t = I_{ph} - I_{d1} - I_{d2} - I_{sh} \quad (5.5)$$

donde  $I_t$  es la corriente en el terminal de medición,  $I_{ph}$  es la corriente foto-generada, las corrientes de los diodos se representan por  $I_{d1}$  e  $I_{d2}$  respectivamente, mientras que  $I_{sh}$  es la corriente del resistor de derivación. Para modelar correctamente la celda solar, se emplea la ecuación de Shockley [193], de esta forma se puede reescribir la ecuación (5.5) de la siguiente manera:

$$I_t = I_{ph} - I_{sd1} \left[ \exp \left( \frac{q(V_t + R_s \cdot I_t)}{n_1 \cdot k \cdot T} \right) - 1 \right] - \dots \quad (5.6)$$

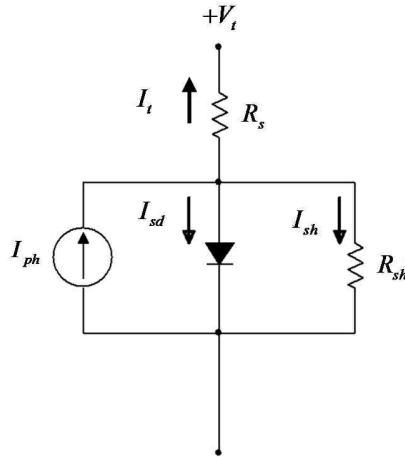
$$I_{sd2} \left[ \exp \left( \frac{q(V_t + R_s \cdot I_t)}{n_2 \cdot k \cdot T} \right) - 1 \right] - \frac{V_t + R_s \cdot I_t}{R_{sh}}$$

donde  $I_{sd1}$  e  $I_{sd2}$  son las corrientes de difusión y saturación, respectivamente.  $V_t$  es el voltaje en el terminal, mientras que las resistencias en serie y la de derivación se representan con  $R_s$  y  $R_{sh}$ . De acuerdo a la ecuación del diodo de Shockley,  $q = 1.602 \times 10^{-19}$  (coulomb) es la magnitud de carga de un electrón, la constante de Boltzmann se representa por  $k = 1.380 \times 10^{-23}$  ( $J/^\circ K$ ),  $n_1$  y  $n_2$ , son los factores de difusión y recombinación,

respectivamente. Por último,  $T$  es la temperatura de la celda en grados Kelvin ( $^{\circ}K$ ). Por lo tanto, la ecuación (5.6) tiene siete parámetros desconocidos:  $R_s$ ,  $R_{sh}$ ,  $I_{ph}$ ,  $I_{sd1}$ ,  $I_{sd2}$ ,  $n_1$  y  $n_2$ . Una precisa identificación de tales parámetros, permite obtener un rendimiento óptimo de la celda solar, por este motivo el proceso de estimación es una tarea de suma importancia.

### 5.2.2 Modelo de diodo simple

En las celdas solares, las corrientes de difusión ( $I_{sd1}$ ) y de saturación ( $I_{sd2}$ ) son diferentes e independientes. En el modelo de diodo simple, ambas corrientes se combinan usando un factor de idealidad no-físico ( $n$ ) [89, 91, 98]. Este modelo, que se muestra en la Figura 5.2, es ampliamente usado para modelar celdas solares, debido a su simplicidad. A diferencia del modelo de doble diodo, este modelo, sólo cuenta con cinco parámetros que deben ser identificados.



**Figura 5.2:** Modelo de diodo simple para las celdas solares.

En base al modelo de diodo simple, la ecuación (5.6) se reduce de la siguiente forma:

$$I_t = I_{ph} - I_{sd} \left[ \exp \left( \frac{q(V_t + R_s \cdot I_t)}{n \cdot k \cdot T} \right) - 1 \right] - \frac{V_t + R_s \cdot I_t}{R_{sh}} \quad (5.7)$$

Como consecuencia de esto, los parámetros que deben ser identificados son:  $R_s$ ,  $R_{sh}$ ,  $I_{ph}$ ,  $I_{sd}$  y  $n$ . En la Tabla 5.1 se presentan los rangos de valores en los que cada

parámetro puede encontrarse, es importante notar que tales rangos se usan para ambos modelos.

Parámetro	Valor inferior	Valor superior
$R_s (\Omega)$	0	0.5
$R_{sh} (\Omega)$	0	100
$I_{ph} (A)$	0	1
$I_{sd} (\mu A)$	0	1
$n$	1	2

**Tabla 5.1:** Rangos de los parámetros de las celdas solares

### 5.2.3 Identificación de parámetros de celdas solares, como un problema de optimización

El problema de modelar celdas solares consiste en encontrar con precisión los parámetros de las ecuaciones (5.6) y (5.7). En el enfoque propuesto, el problema de identificación de parámetros se considera como un problema de optimización para esto se realiza una búsqueda del conjunto de parámetros que produzca la mejor aproximación, para las mediciones de la relación Corriente-Voltaje ( $I$ - $V$ ) obtenidas en celdas solares reales. Por lo tanto, es necesario definir una función objetivo que evalúe la calidad de coincidencia entre el conjunto de parámetros candidatos y los datos experimentales. En esta investigación se consideran tanto el modelo de diodo doble como el de diodo simple para el modelado de celdas solares. En base a esto, se deben reescribir las ecuaciones (5.6) y (5.7), para que puedan reflejar la diferencia entre los valores propuestos y los datos experimentales. De este modo, la función de error para el modelo de diodo doble se define como:

$$f_{DD}(V_t, I_t, \mathbf{x}) = I_t - I_{ph} + I_{sd1} \left[ \exp \left( \frac{q(V_t + R_s \cdot I_t)}{n_1 \cdot k \cdot T} \right) - 1 \right] + \dots \quad (5.8)$$

$$I_{sd2} \left[ \exp \left( \frac{q(V_t + R_s \cdot I_t)}{n_2 \cdot k \cdot T} \right) - 1 \right] + \frac{V_t + R_s \cdot I_t}{R_{sh}}$$

mientras que para el modelo de diodo simple se formula la ecuación (5.9).

$$f_{SD}(V_t, I_t, \mathbf{x}) = I_t - I_{ph} + I_{sd} \left[ \exp \left( \frac{q(V_t + R_s \cdot I_t)}{n \cdot k \cdot T} \right) - 1 \right] + \frac{V_t + R_s \cdot I_t}{R_{sh}} \quad (5.9)$$

En ambas funciones ( $f_{DD}$  y  $f_{SD}$ ) los valores de  $V_t$  e  $I_t$  son obtenidos experimentalmente a partir de una celda solar real;  $\mathbf{x}$  es un vector que contiene los parámetros del modelo, donde  $\mathbf{x} = [R_s, R_{sh}, I_{ph}, I_{sd1}, I_{sd2}, n_1, n_2]$  para el modelo de diodo doble y  $\mathbf{x} = [R_s, R_{sh}, I_{ph}, I_{sd}, n]$  para el modelo de diodo simple. Las ecuaciones (5.8) y (5.9) permiten evaluar la calidad del conjunto de parámetros del modelo  $\mathbf{x}$ , midiendo la diferencia entre el valor real de  $I_t$  y el calculado por el modelo. Por lo tanto, la estimación de parámetros es un proceso que minimiza la diferencia entre los datos medidos y la corriente calculada, por medio del ajuste de los parámetros del modelo. Considerando que la cantidad de datos experimentales es  $N$ , la función objetivo se puede formular por medio del error cuadrático medio (*root mean square error*, RMSE):

$$RMSE(\mathbf{x}) = \sqrt{\frac{1}{N} \sum_{c=1}^N \left( f_M^c(V_t^c, I_t^c, \mathbf{x}) \right)^2} \quad (5.10)$$

donde  $M$  es el tipo de modelo de diodo, doble o simple. La función objetivo se construye por medio del uso de los datos experimentales, que además de ser obtenidos por medio de mediciones, se pueden extraer de las hojas de especificaciones que los fabricantes proveen. Ya que los datos experimentales su?

elen contener ruido, debido al proceso en que éstos son adquiridos, la función objetivo tiende a presentar características altamente multimodales y ruidosas [194, 195]. Bajo estas circunstancias, la mayoría de los enfoques metaheurísticos suelen presentar un mal rendimiento [196].

### 5.3 Identificación de parámetros de celdas solares usando el algoritmo ABC

En esta sección se describe el enfoque propuesto para la identificación y modelado de celdas solares usando el algoritmo ABC. Primero se plantea el problema, es aquí donde se analizan las restricciones del mismo. A continuación, se describe el algoritmo propuesto, para ambos modelos de celdas solares.

### 5.3.1 Planteamiento del problema

El método propuesto codifica los parámetros de la celda solar como una fuente de alimento o solución candidata. La presentación de esta solución depende del tipo de modelo de celda solar: doble diodo o diodo simple. Por lo tanto, cada fuente de alimento usa siete elementos para el modelo de diodo doble y cinco para el modelo de diodo simple, de este modo se definen las variables de decisión dentro del contexto de optimización. Entonces, la tarea de estimación de parámetros puede plantearse como un problema de optimización como sigue:

$$\begin{aligned}
 &\text{Minimizar} && RMSE(\mathbf{X}), \quad \mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_p}], \\
 &&& \mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}], \quad d \in [5, 7], \\
 &\text{Sujeto a} && d = 5 \text{ (SD)} && d = 7 \text{ (DD)} \\
 &&& 0 \leq x_{i,1} (R_s) \leq 0.5 && 0 \leq x_{i,1} (R_s) \leq 0.5 \\
 &&& 0 \leq x_{i,2} (R_{sh}) \leq 100 && 0 \leq x_{i,2} (R_{sh}) \leq 100 \\
 &&& 0 \leq x_{i,3} (I_{ph}) \leq 1 && 0 \leq x_{i,3} (I_{ph}) \leq 1 \\
 &&& 0 \leq x_{i,4} (I_{sd}) \leq 1 && 0 \leq x_{i,4} (I_{sd1}) \leq 1 \\
 &&& 1 \leq x_{i,5} (n) \leq 2 && 0 \leq x_{i,5} (I_{sd2}) \leq 1 \\
 &&& && 1 \leq x_{i,6} (n_1) \leq 2 \\
 &&& && 1 \leq x_{i,7} (n_2) \leq 2
 \end{aligned} \tag{5.11}$$

donde  $N_p$  y  $d$  corresponden al tamaño de la población y al número de dimensiones del problema respectivamente. Aquí las restricciones a las que el RMSE está sujeto corresponden al rango de valores en los que se puede encontrar cada uno de los parámetros del modelo de celda solar que se pretende optimizar.

### 5.3.2 Enfoque computacional para el diseño de celdas solares

El algoritmo propuesto ha sido implementado considerando los dos modelos de celdas solares basados en diodos, mientras que su eficiencia es evaluada usando el criterio de RMSE. Se emplea el algoritmo de optimización ABC para resolver el problema de identificación de parámetros definido en la ecuación (5.11). El procedimiento computacional del enfoque propuesto se puede resumir de acuerdo con los pasos descritos en el Algoritmo 5.2.

**Algoritmo 5.2: Diseño de celdas solares usando ABC**


---

```

1: Leer los  $N$  valores de los datos experimentales de  $V_i$  e  $I_i$ 
2: Almacenar los  $N$  valores en  $\mathbf{ED} = [ED_1, ED_2, \dots, ED_N]$ ,  $ED_i = [V_i, I_i]$ 
3: Inicializar los parámetros de ABC:  $N_p$ ,  $d = 5$  o  $d = 7$ ,  $max\_ciclo$ ,  $limite$ 
4: Generar la población inicial  $\mathbf{X}$  de  $N_p$  soluciones candidatas, con  $d$  dimensiones
5: Calcular  $RMSE(\mathbf{X})$ 
6:  $ciclo \leftarrow 1$ 
7: while  $ciclo \leq max\_ciclo$  do
8:   for  $i = 1$  to  $N_p/2$  do
9:     Producir una nueva solución  $B_i$  usando la ecuación (5.2)
10:    Calcular  $fit_i$ 
11:    Aplicar el proceso de selección voraz
12:   endfor
13:   Calcular  $Prob_i$ 
14:   for  $i = 1$  to  $N_p/2$  do
15:     Seleccionar una nueva solución dependiendo de  $Prob_i$ 
16:     Producir una nueva solución  $B_i$ 
17:     Calcular  $fit_i$ 
18:     Aplicar el proceso de selección voraz
19:   endfor
20:   if (una solución es abandonada) then
21:     Generar una nueva solución usando la ecuación (5.1)
22:   endif
23:   Almacenar la mejor solución actual
24:    $ciclo \leftarrow ciclo + 1$ 
25: endwhile

```

---

## 5.4 Resultados experimentales

Con el objetivo de determinar el rendimiento del enfoque propuesto, se ha probado el algoritmo usando una celda solar de silicio de la compañía R. T. C. de Francia, de la cual se extraen los valores experimentales bajo condiciones estándar, mejor conocidas como STC. La celda tiene un diámetro de 57mm y trabaja bajo las siguientes condiciones: 1 *sun* ( $100W/m^2$ ) a una temperatura de  $T = 33^\circ C$ , sin embargo, para probar el rendimiento del método basado en ABC, se han incluido cuatro valores de temperatura:  $T = 25^\circ C$ ,  $T = 50^\circ C$ ,  $T = 75^\circ C$ ,  $T = 100^\circ C$ . En esta sección, se presentan dos grupos diferentes de resultados. En la primera parte el método propuesto se emplea para extraer los parámetros de la celda usando ambos modelos de diodo. Finalmente, en la segunda parte se compara el resultado del enfoque basado en ABC con otros enfoques similares existentes en la literatura.



Para la realización de los experimentos, el algoritmo ABC se ha configurado considerando los parámetros que se muestran en la Tabla 5.2. Una vez que dichos valores se determinan mediante experimentos exhaustivos, son usados para todos los experimentos.

$max\_ciclo$	$N_p$	$limite$
10000	150	$N_p \cdot d$

**Tabla 5.2:** Parámetros de configuración del algoritmo ABC.

El parámetro *límite* se calcula como el producto de  $N_p \cdot d$ , como ya se mencionó anteriormente  $N_p$  es el tamaño de la población y  $d$  la dimensión del problema. Para los experimentos realizados, el criterio de parada es el número máximo de iteraciones o ciclos  $max\_ciclo$ . Sin embargo, si el valor de la función objetivo, para la mejor solución candidata, se mantiene sin cambio durante el 10% del valor máximo de iteraciones, entonces el algoritmo se detiene.

#### 5.4.1 Resultados experimentales del algoritmo ABC

Este experimento, presenta los resultados del enfoque propuesto cuando se emplea para extraer los parámetros de las celdas solares, considerando los modelos de doble diodo y de diodo simple. Para este fin, se han recolectado 26 mediciones de una celda solar física, estas muestras se presentan en la Tabla 5.4, habiéndose seleccionado para mantener la compatibilidad con trabajos similares existentes en la literatura. Los parámetros obtenidos por el método basado en ABC para los modelos de doble diodo y de diodo simple se presentan en la Tabla 5.3. Como el modelo de diodo simple tiene cinco parámetros y el de doble diodo siete, en la Tabla 5.3 existen algunos valores que no se calculan para uno u otro modelo según corresponda, la inexistencia de tales parámetros se representa por el símbolo (-).

Para evaluar la precisión del modelo identificado, se emplean cuatro índices diferentes de rendimiento: el error cuadrático medio normalizado (NMRSE), el error relativo  $R_{error}$ , el error absoluto medio (MAE), el error absoluto medio normalizado (NMAE) y el error de sesgo medio (MBE).

Parámetro	Doble diodo	Diodo Simple
$R_s (\Omega)$	0.0364	0.0364
$R_{sh} (\Omega)$	53.7804	53.6433
$I_{ph} (A)$	0.7608	0.7608
$I_{sd} (\mu A)$	-	32.51
$I_{sd1} (\mu A)$	0.0407	-
$I_{sd2} (\mu A)$	0.2874	-
$n_1$	1.4495	-
$n_2$	1.4885	-
$n$	-	1.4817
RMSE	9.861 E-04	9.862 E-04

**Tabla 5.3:** Parámetros obtenidos por el método basado en ABC para los modelos de doble diodo y de diodo simple.

Matemáticamente, el error cuadrático medio normalizado se define como:

$$NRMSE(\mathbf{x}) = \frac{RMSE}{\max(I_{t-calculated}) - \min(I_{t-calculated})} \quad (5.12)$$

Donde  $\min(I_{t-calculated})$  y  $\max(I_{t-calculated})$  son los valores mínimo y máximo de  $I_{t-calculated}$  sobre las  $N$  muestras. Los valores máximo y mínimo se usan también para la normalización del MAE y el MBE. El error relativo  $R_{error}$  evalúa la diferencia entre la corriente medida  $I_{t-measured}$  y la corriente calculada  $I_{t-calculated}$  por el algoritmo basado en ABC, para cada modelo de celda solar y se obtiene por medio de la siguiente ecuación:

$$R_{error} = \frac{I_{t-measured} - I_{t-calculated}}{I_{t-measured}} \times 100 \quad (5.13)$$

El error absoluto medio (MAE) y su respectivo valor normalizado (NMAE) se calculan usando las ecuaciones (5.14) y (5.15), en ellas el valor de  $N$  corresponde a la cantidad de datos experimentales que se emplean, en este caso  $N=26$ .

$$MAE = \sum_{i=1}^N \frac{|I_{t-measured} - I_{t-calculated}|}{N} \quad (5.14)$$

$$NMAE = \sum_{i=1}^N \frac{|I_{t-measured} - I_{t-calculated}| / I_{t-measured}}{N} \quad (5.15)$$

Otra métrica de error usada es el error de sesgo (MBE) medio y su respectivo valor normalizado (NMBE), que se definen como:

$$MBE = \sum_{i=1}^N \frac{(I_{t-\text{measured}} - I_{t-\text{calculated}})}{N} \quad (5.16)$$

$$NMBE = \frac{MBE}{\max(I_{t-\text{calculated}}) - \min(I_{t-\text{calculated}})} \quad (5.17)$$

La Tabla 5.4 presenta los resultados de este experimento, tales resultados incluyen los datos experimentales  $V_t$  e  $I_t$ , los resultados obtenidos por el método propuesto  $I_{t-\text{calculated}}$ , los respectivos valores del error relativo  $R_{\text{error}}$  y sus valores normalizados, todo ello para ambos modelos de celda solar.

Datos	$V_t (V)$ Medido	$I_t (A)$ Medido	$I_{t-\text{calculated}} (A)$ ABC modelo de doble diodo	$R_{\text{error}}$ ABC modelo de doble diodo	$NR_{\text{error}} (\%)$ ABC modelo de doble diodo	$I_{t-\text{calculated}} (A)$ ABC modelo de doble diodo	$R_{\text{error}}$ ABC modelo de doble diodo	$NR_{\text{error}} (\%)$ ABC modelo de doble diodo
1	-0.2057	0.7640	0.7640	-9.2908 E-05	36.8310	0.7641	-0.0001	36.6608
2	-0.1291	0.7620	0.7626	-0.0006	22.8277	0.7626	-0.0006	22.6153
3	-0.0588	0.7605	0.7613	-0.0008	18.1264	0.7613	-0.0008	17.95165
4	0.0057	0.7605	0.7601	0.0003	47.2470	0.7601	0.0003	47.4202
5	0.0646	0.7600	0.7590	0.0009	61.7485	0.7590	0.0009	62.1258
6	0.1185	0.7590	0.7580	0.0009	62.0349	0.7580	0.0009	62.4765
7	0.1678	0.7570	0.7571	-0.0001	36.5502	0.7571	-0.0001	36.8073
8	0.2132	0.7570	0.7561	0.0008	59.6029	0.7561	0.0008	60.1225
9	0.2545	0.7555	0.7550	0.0004	48.8006	0.7550	0.0004	49.2543
10	0.2924	0.7540	0.7536	0.0003	46.9719	0.7536	0.0003	47.4304
11	0.3269	0.7505	0.7513	-0.0008	17.2776	0.7513	-0.0008	17.4642
12	0.3585	0.7465	0.7473	-0.0008	18.3219	0.7473	-0.0008	18.5101
13	0.3873	0.7385	0.7401	-0.0016	0	0.7401	-0.0016	0
14	0.4137	0.7280	0.7273	0.0006	54.4874	0.7273	0.0006	54.9849
15	0.4373	0.7065	0.7069	-0.0004	28.2650	0.7069	-0.0004	28.5214
16	0.4590	0.6755	0.6752	0.0002	45.2437	0.6752	0.0002	45.6933
17	0.4784	0.6320	0.6307	0.0012	70.0782	0.6307	0.0012	70.8228
18	0.4960	0.5730	0.5718	0.0011	65.8199	0.5718	0.0011	66.6002
19	0.5119	0.4990	0.4995	-0.0005	24.8089	0.4995	-0.0005	25.2670
20	0.5265	0.4130	0.4136	-0.0006	23.4270	0.4136	-0.0006	23.8922
21	0.5398	0.3165	0.3175	-0.0010	14.2999	0.3175	-0.0010	14.6298
22	0.5521	0.2120	0.2121	-0.0001	34.7889	0.2121	-0.0001	35.1718
23	0.5633	0.1035	0.1022	0.0012	68.7767	0.1022	0.0012	69.2431
24	0.5736	-0.0100	-0.0087	-0.0012	7.57006	-0.0086	-0.0013	7.13579
25	0.5833	-0.1230	-0.1255	0.0025	100	-0.1254	0.0024	100
26	0.5900	-0.2100	-0.2085	-0.0014	2.6948	-0.2084	-0.0015	1.41935

**Tabla 5.4:** Mediciones en la terminal de la celda solar, valores estimados por el ABC, error relativo y error relativo normalizado para los modelos de celda solar de doble diodo y diodo simple.

En la Tabla 5.5 se muestran los valores del RMSE, el RMSE normalizado (NRMSE), MAE, NMAE, MBE y el NMBE para el método propuesto basado en ABC.

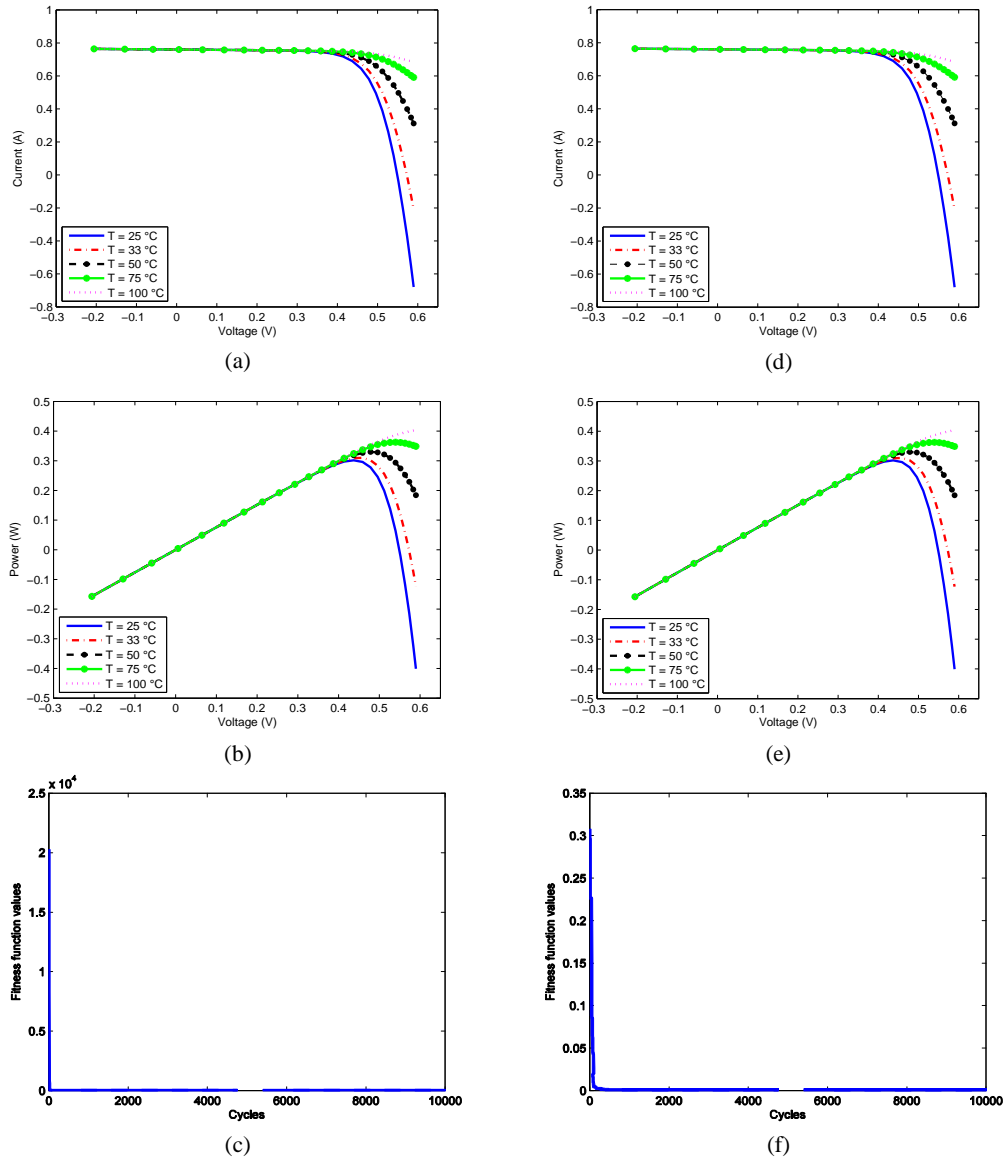
Modelo	RMSE	NRMSE (%)	MAE	NMAE (%)	MBE
Doble diodo	9.8619 E-04	62.53	8.2918 E-04	-0.47	5.8807 E-07
Diodo simple	9.8629 E-04	62.70	8.3034 E-04	-0.49	-1.5448 E-06

**Tabla 5.5:** Índices de rendimiento para los modelos de celdas solares: doble diodo, y diodo simple.

Considerando los parámetros de los modelos de la Tabla 5.4 es posible obtener la potencia de la celda solar ( $P = I \times V$ ). La Figura 5.3 muestra las gráficas de corriente frente al voltaje a las diferentes temperaturas propuesta, la potencia y el valor de la función objetivo para ambos modelos basados en diodo. En ella se muestran los resultados obtenidos por el enfoque propuesto, considerando los dos modelos de celdas solares y cinco valores de temperatura. En dicha figura se puede verificar que la técnica propuesta basada en ABC obtiene mejores modelos, produciendo una aproximación precisa a los datos experimentales. Sin embargo, la influencia de las temperaturas modifica los valores de corriente en el modelo, lo cual resulta evidente. Este hecho afecta directamente a los valores de potencia. Además, la evolución del proceso de optimización da evidencia suficiente de que el método propuesto permite encontrar los modelos apropiados de celda solares en un número de iteraciones relativamente bajo.

#### 5.4.2 Comparaciones con otros enfoques similares

Con el propósito de demostrar el desempeño del enfoque propuesto basado en ABC, se hace una comparación de los resultados obtenidos con otras implementaciones similares, que ya han sido reportadas en la literatura para dar solución al mismo problema. Los métodos usados para esta comparación usan como método de optimización los siguiente algoritmos: búsqueda de armonías (*Harmony search*, HS) [98], optimización por enjambre de partículas (*ParticleSwarm optimization*, PSO) [97], algoritmos genéticos (*Genetic Algorithm*, GA) [102] y búsqueda de alimento bacteriano (*Bacterial Foraging*, BFA) [170]. Para llevar a cabo la comparación, todos los algoritmos se ejecutan 35 veces, con esto, se calcula el promedio del RMSE, la media y la desviación estándar (STD). En las Tablas 5.6 y 5.7 se presentan los resultados obtenidos mediante este análisis, para el modelo de doble diodo y el de diodo simple, respectivamente.



**Figura 5.3:** Para el modelo de diodo doble: (a) voltaje medido frente a corriente calculada por ABC para diferentes valores de temperatura, (b) voltaje medido frente a potencia obtenida por ABC a diferentes temperaturas, (c) evolución del RMSE. Para el modelo diodo simple: (d) voltaje medido frente a corriente calculada por ABC para diferentes valores de temperatura, (e) voltaje medido frente a potencia obtenida por ABC a diferentes temperaturas, (f) evolución del RMSE.

La Tabla 5.6 presenta el análisis comparativo para el modelo de doble diodo, a partir de estos resultados, es posible observar que la técnica basada en ABC tiene un mejor desempeño. El valor de PRMSE corresponde al promedio del RMSE después de 35 experimentos independientes (o ejecuciones). El valor de STD puede interpretarse como un índice de estabilidad, el cual refleja la capacidad del algoritmo para producir el mismo

resultado cuando se ejecuta en múltiples ocasiones. Siguiendo con esta metodología, la Tabla 5.7 muestra los resultados del análisis comparativo para el modelo de diodo simple. Los resultados evidencian que el algoritmo propuesto es más eficiente en comparación con los algoritmos HS, PSO, GA y BFA, en términos de los valores de RMSE y STD.

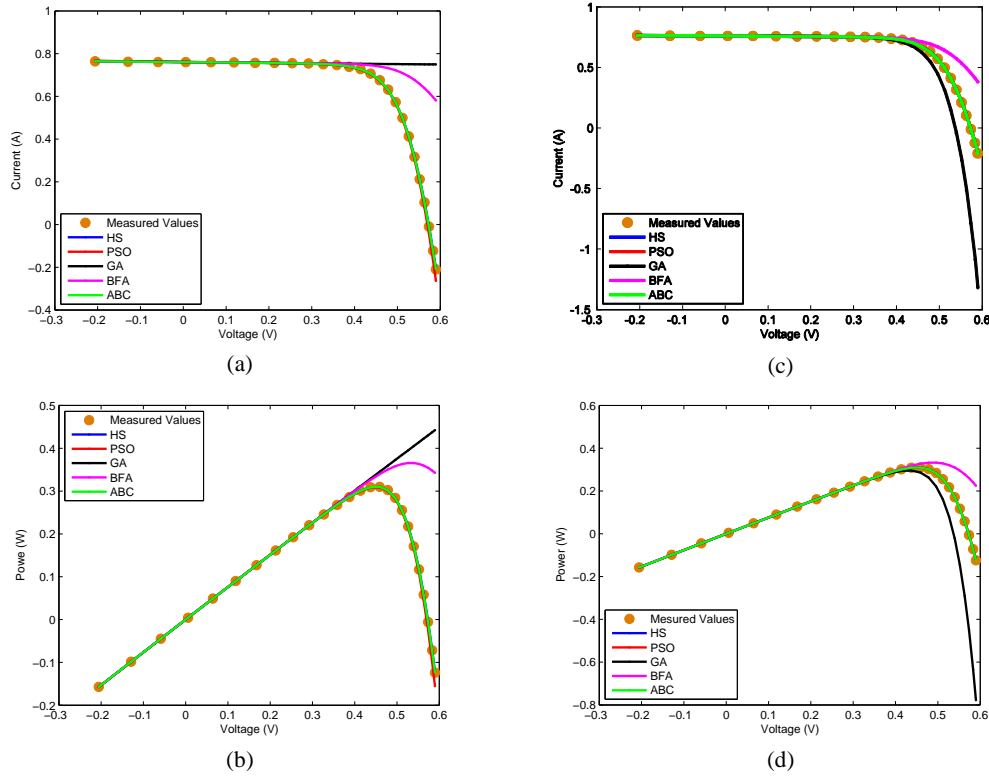
Parámetro	ABC	HS	PSO	GA	BFA
$R_s (\Omega)$	0.0364	0.0354	0.0325	0.0364	0.0351
$R_{sh} (\Omega)$	53.7804	46.8269	43.1034	53.7185	60.0000
$I_{ph} (A)$	0.7608	0.7617	0.7623	0.7608	0.7609
$I_{sd1} (\mu A)$	0.0407	0.1245	0.4767	0.0001	0.0094
$I_{sd2} (\mu A)$	0.2874	0.2547	0.0100	0.0001	0.0453
$n_1$	1.4495	1.4943	1.5172	1.3355	1.3809
$n_2$	1.4885	1.4998	2.0000	1.4810	1.5255
<b>RMSE</b>	9.861 E-04	0.0013	0.0166	0.3604	0.0012
<b>PRMSE</b>	0.0010	0.0683	0.0715	0.0229	0.0245
<b>STD</b>	3.285 E-05	0.5111	0.3109	0.0199	0.3697

**Tabla 5.6:** Comparación de los resultados para el modelo de doble diodo.

Parámetro	ABC	HS	PSO	GA	BFA
$R_s (\Omega)$	0.0364	0.0366	0.0354	0.0299	0.0325
$R_{sh} (\Omega)$	53.6433	53.5946	59.0120	42.3729	50.8691
$I_{ph} (A)$	0.7608	0.7607	0.7607	0.7619	0.7602
$I_{sd} (\mu A)$	0.3251	0.3049	0.4000	0.8087	0.8000
$n$	1.4817	1.4753	1.5033	1.5751	1.6951
<b>RMSE</b>	9.862 E-04	9.510 E-04	0.0013	0.0190	0.029
<b>PRMSE</b>	0.0010	0.0039	0.2544	0.0551	0.0152
<b>STD</b>	1.497 E-05	0.7268	0.0289	0.0735	0.0586

**Tabla 5.7:** Comparación de los resultados para el modelo de diodo simple.

En la Figura 5.4 se muestran las gráficas de los valores del voltaje medido frente a la corriente obtenida por cada una de las técnicas comparadas, y del mismo modo se presenta la potencia generada. Con fines experimentales, solamente se emplea la temperatura  $T = 33^\circ C$  para el análisis comparativo de los resultados.



**Figura 5.4:** Comparación de los métodos ABC, HS, PSO, GA y BFA para el modelo de doble diodo en (a) voltaje medido frente a la corriente calculada y (b) voltaje medido frente a la potencia y para el modelo de diodo simple en (c) voltaje medido frente a la corriente calculada, (d) voltaje medido frente a la potencia.

A partir de los resultados mostrados en la Figura 5.4, se puede observar cómo los métodos tales como GA y BFA no logran estimar los parámetros que se ajusten a las muestras medidas. Por otro lado, a pesar de que PSO y HS a simple vista pareciera que logran valores similares a los medidos, las Tablas 5.6 y 5.7 demuestran la falta de precisión de los resultados. De este modo el algoritmo basado en ABC es el que proporciona el mejor desempeño.

## 5.5 Resumen

En este capítulo se presenta el algoritmo de optimización basado en enjambre de abejas ABC. Este método consta de diferentes etapas que colaboran con la exploración y explotación del espacio de búsqueda que se pretende optimizar. Tales etapas y la descripción completa del algoritmo son analizadas y detalladas.

Por otra parte, para comprobar la eficiencia de esta técnica de optimización, se propone su implementación para dar solución al problema de diseño de celdas solares. Para esto se emplean dos modelos: doble diodo y diodo simple. En esta implementación el algoritmo ABC se encarga de buscar los parámetros del modelo de la celda solar que proporciona el mejor resultado, de acuerdo al error cuadrático medio (RMSE). Los resultados experimentales, muestran diferentes métricas que respaldan la capacidad de este método para estimar los parámetros de las celdas solares. Del mismo modo, se comparó el algoritmo propuesto con otros métodos similares propuestos en la literatura con lo cual se verifica y comprueba que el algoritmo ABC es capaz de proporcionar mejores resultados para el problema planteado.





# Capítulo 6

## Conclusiones y trabajo futuro

---

Este capítulo finaliza con las conclusiones más relevantes en relación a la investigación desarrollada y reflejada en la presente memoria de tesis doctoral. Sin embargo, aún existe un gran campo de acción para los métodos analizados y los problemas estudiados. De este modo, se presentan las conclusiones, las cuales, de acuerdo a la metodología que se ha seguido, se dividen en dos: conclusiones específicas y conclusiones generales. Dentro de las específicas, se realiza una revisión de los temas descritos para cada uno de los problemas tratados y de cómo se abordaron sus soluciones. Mientras que en las generales se describen de forma genérica, las experiencias obtenidas a lo largo de todo el proceso que involucró la realización de este trabajo de investigación. Finalmente se describe el trabajo futuro referente a los algoritmos metaheurísticos y sus posibles mejoras e implementaciones.

### 6.1 Conclusiones específicas

En esta sección se incluyen los aspectos más relevantes relativos a las implementaciones o modificaciones de los algoritmos metaheurísticos analizados en los capítulos anteriores. Primeramente se analiza la modificación del algoritmo EMO empleando

la regla de aprendizaje OBL, que se describió en la sección 3.3. Como segundo punto, se incluyen las conclusiones referentes a la implementación del algoritmo EMO para el modelado de filtros digitales (sección 3.5). Como siguiente punto, se concluye el tema referente a la detección de glóbulos blancos en imágenes médicas usando EMO el cual se expuso en la sección 3.7. Para finalizar con los temas referentes a las modificaciones e implementaciones de EMO se realiza un análisis de lo expuesto en la sección 3.8 referente a la localización y detección de objetos mediante plantillas.

Por otro lado, todo lo expuesto referente a los temas de umbralización (Capítulo 4) y celdas solares (Capítulo 5) también es analizado y concluido en esta sección.

### 6.1.1 *Opposition-Based Electromagnetism-Like Optimization (OBEMO)*

En la sección 3.3 se propuso el algoritmo OBEMO, el cual combina la estrategia de aprendizaje (OBL) basada en lo opuesto con la técnica de optimización EMO estándar. Como ya se mencionó, OBL es una técnica de aprendizaje automático, la cual considera al mismo tiempo la solución estimada actual y su valor opuesto, para obtener una rápida aproximación para una solución candidata dada. El EMO estándar se ha mejorado usando dos pasos de OBL, concretamente en la inicialización de la población y para producir nuevas poblaciones. El algoritmo mejorado reduce significativamente el esfuerzo de cómputo requerido, evitando demeritar alguna de las características de EMO.

Se emplea un conjunto de 14 funciones de referencia para realizar un estudio experimental. Los resultados son respaldados por una prueba estadísticamente significativa (prueba de Wilcoxon), lo que demuestra que OBEMO es igual de preciso que EMO, pero requiere un número reducido de iteraciones. De forma similar, el método propuesto es más rápido que otros algoritmos basados en EMO que se analizan en el estado del arte, como son HEMO [14] y FEMO [105], cabe señalar que a pesar de su rapidez mantiene la precisión.

### 6.1.2 *Modelado de filtros digitales usando el algoritmo EMO*

Dentro de las implementaciones del algoritmo EMO, en la sección 3.5 se presentó un nuevo algoritmo para la identificación y modelado de filtros IIR, el cual está basado en EMO. Bajo este enfoque, la adaptación de los coeficientes es considerada como un problema de optimización, para la cual se utiliza como función objetivo el error cuadrático medio (MSE) existente entre la respuesta del sistema desconocido y la estimada por el filtro durante cierto periodo de tiempo. Guiados por los valores de la función objetivo, el conjunto de coeficientes

que modelan el filtro se obtienen usando el algoritmo EMO hasta encontrar la mejor configuración posible (mínimo global).

El desempeño del algoritmo propuesto, fue comparado experimentalmente frente al algoritmo de identificación IIR basado en la técnica PSO [57] usando un conjunto de pruebas estándar [177]. De los resultados se desprende que el algoritmo basado en EMO, entrega resultados más precisos en prácticamente el mismo número de iteraciones.

Aunque el algoritmo propuesto ofrece evidencia de un mejor desempeño, el objetivo de la investigación tiene como propósito mostrar que los sistemas basados en principios físicos como el *Electromagnetism-Like*, se pueden considerar como una alternativa sumamente atractiva para la identificación de parámetros en los filtros IIR.

### 6.1.3 Detector de glóbulos blancos usando el algoritmo EMO

Para comprobar el rendimiento del algoritmo EMO se propone una implementación dentro del área de procesamiento de imágenes digitales. En la sección 3.7, se introdujo un algoritmo para la detección automática de glóbulos blancos que están contenidos dentro de imágenes complejas de frotis sanguíneos, que considera este proceso como un problema de detección de formas circulares. El enfoque propuesto está basado en la técnica metaheurística conocida como EMO (*electromagnetism-like optimization*), la cual emplea los principios del electromagnetismo para dar solución a los problemas de optimización complejos. El método propuesto usa la codificación de tres puntos de borde no-colineales como círculos candidatos, los cuales se prueban en el mapa de bordes de la imagen que contiene las células sanguíneas. De esta investigación, se deriva una nueva función objetivo, la cual se propone para medir la similitud entre un círculo candidato y un glóbulo blanco, basándose en la información del mapa de bordes obtenido tras segmentar la imagen. Usando como referencia los valores de la función objetivo, el conjunto de círculos candidatos codificados (partículas cargadas) son evolucionados usando el algoritmo EMO, para que puedan coincidir con las células sanguíneas contenidas en el mapa de bordes.

El rendimiento y efectividad del método propuesto basado en EMO, ha sido comparado con otros detectores de glóbulos blancos propuestos en la literatura, como son: el enfoque basado en vectores soporte de contorno [73], el método iterativo de Otsu [74], el algoritmo de Wang [75] y el detector basado en algoritmos genéticos [76]. Dichos métodos se compararon considerando un amplio conjunto de imágenes, con diferentes niveles de

complejidad. Los resultados experimentales demuestran el gran desempeño de la técnica propuesta en términos de precisión, robustez y estabilidad.

#### 6.1.4 *Localización y detección de objetos mediante el uso de plantillas empleando EMO como estrategia de búsqueda*

Para terminar las implementaciones y mejoras referentes al algoritmo EMO, en la sección 3.10 se propone un nuevo método que reduce el número de localidades de búsqueda en el proceso de localización y detección de objetos mediante plantillas. El algoritmo usa una versión mejorada de EMO, la cual incluye una modificación del proceso de búsqueda local para acelerar la explotación del espacio de búsqueda. Esta modificación reduce la cantidad de perturbaciones alrededor de cada partícula a sólo un grupo pequeño de muestras aleatorias. Como resultado, el nuevo algoritmo EMO puede reducir el número de evaluaciones de la función objetivo preservando las capacidades de búsqueda del EMO original.

En el enfoque propuesto, las partículas representan las posiciones de búsqueda, las cuales se mueven a lo largo de las posiciones de la imagen fuente. Los valores de los coeficientes de correlación cruzada normalizada (NCC) se usan como función objetivo y a su vez influyen en la carga ejercida entre las partículas. El NCC evalúa la calidad de la coincidencia que existe entre la imagen plantilla y la región coincidente en la imagen fuente, para una posición dada (partícula). La cantidad de evaluaciones de NCC se reduce también considerando una memoria en la que se almacenan estos valores, para las posiciones previamente visitadas, con esto se evita re-evaluar las mismas partículas. Tomando como guía los valores de la función objetivo, las posiciones candidatas son modificadas por los operadores EMO en cada iteración, esto se repite hasta que se encuentra la mejor semejanza. El método propuesto alcanza el mejor balance sobre los métodos empleados para este problema, en términos de precisión y costo computacional.

Por otro lado, se realizó la comparación del rendimiento del algoritmo propuesto con otros métodos similares, considerando diferentes imágenes, las cuales presentan una gran variedad de formatos y grados de complejidad. Los resultados experimentales demuestran el gran desempeño de la técnica propuesta en términos de precisión y cantidad de evaluaciones del NCC.

### 6.1.5 Segmentación multinivel de imágenes digitales usando Harmony Search

En el capítulo 4 se presentó un método de segmentación multinivel basado en el algoritmo original de búsqueda de armonías (*harmony search*, HSA). El enfoque combina las capacidades del algoritmo HSA y el uso de las funciones objetivo propuestas en los métodos de segmentación multinivel de Otsu y Kapur. Para medir el desempeño del enfoque propuesto, se usa la relación señal a ruido de pico (PSNR) que además sirve como métrica de calidad de la segmentación, considerando las coincidencias entre la imagen original y la imagen segmentada. En esta investigación, se realizó una implementación simple del HSA, es decir, no se considera ninguna modificación del mismo, porque con esto se demuestra la efectividad del método de optimización aplicado en tareas de procesamiento de imágenes.

El estudio realizado, explora la comparación entre dos versiones del algoritmo de segmentación basado en HSA: una de estas versiones usa como función objetivo el método de Otsu, mientras que la otra versión emplea el criterio de Kapur. Los resultados expuestos en el capítulo 4, muestran que la función de Otsu provee mejores resultados en comparación al criterio de Kapur. Esta conclusión se comprobó estadísticamente considerando la prueba de Wilcoxon.

El enfoque propuesto se comparó con otras técnicas similares, las cuales se ha implementado mediante diferentes algoritmos de optimización como son: GA, PSO y BF. La eficiencia del algoritmo se evaluó en términos del PSNR y de la desviación estándar. Los resultados experimentales, dan evidencia del sobresaliente desempeño, precisión y convergencia del algoritmo propuesto en comparación con sus contrapartes.

### 6.1.6 Identificación de parámetros de celdas solares usando el algoritmo ABC

En el capítulo 5, se introdujo el uso del algoritmo de colonias artificiales de abejas (*artificial bee colony*, ABC), para estimar con precisión los parámetros de las celdas solares. En este enfoque, el proceso de estimación se considera como un proceso de optimización. De este modo, el método codifica los parámetros de la celda solar como una solución candidata. Una función objetivo se encarga de evaluar la calidad de coincidencia que existe entre una solución candidata y los datos experimentales. Considerando los valores de la función objetivo, el conjunto de soluciones candidatas se modifican por medio de los operadores del algoritmo ABC. El propósito de este proceso, es encontrar los valores que tengan una mayor aproximación a las mediciones de voltaje y corriente extraídas de celdas solares reales.

El algoritmo propuesto basado en ABC se ha comparado con otras técnicas similares propuestas en la literatura como son: HSA, PSO, GA y BFA. De forma similar, se evaluó la eficiencia y robustez del método que emplea ABC. Los resultados experimentales, demuestran un alto desempeño, precisión y convergencia del método propuesto para el problema de diseño de celdas solares. Por lo tanto, en comparación con otros métodos similares existentes en la literatura, el uso del algoritmo ABC resulta una mejor alternativa, ya que permite encontrar los mejores parámetros, empleando ambos modelos de celdas solares basados en diodos.

## 6.2 Conclusiones generales

A lo largo de los capítulos que reflejan el resultado de la investigación realizada, se analizaron diversos problemas dentro de las áreas de procesamiento de señales, procesamiento de imágenes y energías alternativas (ingeniería). Los problemas presentados, fueron seleccionados por dos razones: *a)* la importancia dentro de las áreas antes mencionadas y *b)* su complejidad. Con base en esto, se generaron soluciones, mediante el uso de las técnicas de optimización metaheurísticas. El propósito de los enfoques propuestos, es básicamente crear alternativas efectivas, que den soluciones fiables y robustas a los problemas planteados. Los algoritmos que se usaron son: *electromagnetism-like optimization* (EMO), *harmony search* (HSA) y *artificial bee colony* (ABC). Los cuales se seleccionaron teniendo en cuenta los operadores que emplean y la aceptación que éstos han tenido en la comunidad científica.

Para verificar el rendimiento de la implementación de los algoritmos metaheurísticos en los problemas seleccionados, se desarrollaron diferentes conjuntos de experimentos, en los cuales se prueban las capacidades de los métodos en diversas condiciones de complejidad. Dentro de las pruebas experimentales, se compararon los algoritmos propuestos, con técnicas similares incluidas en el estado del arte de las áreas de aplicación. Los resultados experimentales se analizaron empleando pruebas estadísticas no paramétricas, con esto se comprueba que los valores obtenidos no ocurrieron por casualidad, y por lo tanto son lo suficiente diferentes para ser considerados mejores.

Los resultados experimentales y las comparaciones demuestran que los métodos propuestos son mejores en los términos que cada aplicación requiere (tiempo de computo, evaluaciones, precisión, etc.). El objetivo de esta investigación es generar soluciones alternativas basadas en algoritmos metaheurísticos, las cuales puedan ser suficientemente

competitivas para la resolución de los problemas que se plantean. Logrando con esto demostrar que pueden ser implementados con éxito en aplicaciones que van más allá de las funciones matemáticas de prueba.

Por otra parte, es importante mencionar que como consecuencia de los buenos resultados obtenidos para cada una de las implementaciones, se han logrado publicar varios artículos en revistas internacionales [119 - 124], cumpliendo así con los objetivos planteados al principio de esta investigación.

### 6.3 Trabajo futuro

Hasta este punto se cumplieron los objetivos planteados para esta investigación, sin embargo, dentro de estas líneas de investigación aún quedan muchas vertientes por explorar. Como parte del trabajo futuro, se tiene planeada la creación de un algoritmo metaheurístico, cuyos operadores sean eficientes y sencillos, generando así un método competitivo con un balance entre la exploración y la explotación del espacio de búsqueda.

En el tema procesamiento de imágenes, se continuará analizando las áreas en las cuales es posible aplicar las técnicas de optimización. Una de las que ya se tiene conocimiento es el filtrado en frecuencia de imágenes digitales, donde se requiere encontrar los parámetros óptimos que modelen al filtro deseado. El tema de análisis de imágenes médicas también se tiene contemplado, y a pesar de que en medicina nada está definido completamente, se pretende generar aplicaciones que permitan analizar los globos oculares con el fin de extraer características que permitan la detección de enfermedades como el glaucoma.

Por otra parte, debido a que las energías alternativas son un tópico de interés general, se tiene prevista la implementación de algoritmos metaheurísticos para el modelado de celdas de hidrógeno tipo PEM. Del mismo modo se continúan analizando problemas como es el acomodo de aerogeneradores en granjas eólicas, el cual es un campo de aplicación para los algoritmos de optimización.

Por último, es importante mencionar que aunque estas áreas son de interés actualmente, se tiene la intención de buscar dentro de temas ingeniería, problemas en los cuales estas técnicas puedan dar soluciones efectivas.





# English Summary

---

## 1 Introduction

### 1.1 *Previous work*

Global Optimization [1, 2] has been applied for many areas of science [3], engineering [4], economics [5, 6] and others whose definition requires mathematical modelling [7, 8]. The goal of this research work is the study of appropriate global optimization methods and its use in three specific areas of implementation, namely: 1) digital filter design; 2) image processing and 3) alternative energies. At the same time, improvements of the selected algorithms are presented.

Global optimization aims to find the global optimum for an objective function which has been defined over a given search space. The difficulties associated with the use of mathematical methods over global optimization problems have contributed to the development of alternative solutions. Linear programming and dynamic programming techniques, for example, often have failed in solving (or reaching local optimum at) NP-hard problems which feature a large number of variables and non-linear objective functions. In order to overcome such problems, researchers have proposed metaheuristic-based algorithms for searching near-optimum solutions.

Metaheuristic algorithms are stochastic search methods that mimic the metaphor of biological or physical phenomena. The core of such methods lies on the analysis of collective behavior of relatively simple agents working on decentralized systems. Such systems typically gather an agent's population that can communicate to each other while sharing a common environment. Although a non-centralized control algorithm regulates the agent behavior, the agent can solve complex tasks by analyzing a given global model and harvesting cooperation to other elements. Therefore, a novel global behavior evolves from interaction among agents as it can be seen on typical examples that include ant colonies, animal herding, bird flocking, fish schooling, honey bees, bacteria, charged particles and many more. Some other metaheuristic optimization algorithms have been recently proposed to solve optimization problems, such as:

- Genetic Algorithms (GA) [9] that modifies a solution population using operators that imitates the genetic.
- Particle Swarm Optimization (PSO) [10], which creates a particles population that is modified using operations based on bird flocking or fish school.
- Differential Evolution (DE) [11] that uses different operations of mutation and crossover to create new solutions, which compete with a population until a best solution is found.
- Ant Colony Optimization (ACO) [12], based on the pheromones that ants employ in the process of searching new food sources.

In the group of metaheuristic algorithms, *Electromagnetism-like algorithm* (EMO) is a relatively new population-based method which was firstly introduced by Birbil and Fang [13] to solve continuous optimization models using bounded variables. The algorithm imitates the attraction–repulsion mechanism between charged particles in an electromagnetic field. Each particle represents a solution and carries a certain amount of charge which is proportional to the quality of the solution (objective function). In turn, solutions are defined by position vectors which give real positions for particles within a multi-dimensional space. Moreover, objective function values of particles are calculated considering such position vectors. Each particle exerts repulsion or attraction forces over other population members; the resultant force acting over a particle is used to update its position. Clearly, the idea behind the EMO methodology is to move particles towards the optimum solution by exerting attraction or repulsion forces. Unlike other traditional metaheuristics techniques such as GA, PSO, DE and ACO whose population members

exchange materials or information between each other, the EMO methodology assumes that each particle is influenced by all other particles in the population, mimicking other heuristics methods such as PSO and ACO. Although the EMO algorithm shares some characteristics with PSO and ACO, recent works have exhibited its better accuracy regarding optimal parameters [14-18], yet showing convergence. EMO has been successfully applied to solve different sorts of engineering problems such as flow-shop scheduling [19], vehicle routing [21], array pattern optimization in circuits [22], neural network training [23] and image processing [24].

Another interesting optimization technique is the harmony search algorithm (HSA) introduced by Geem et al. [25], it is an evolutionary optimization algorithm which is based on the metaphor of the improvisation process that occurs when a musician searches for a better state of harmony. The HSA generates a new candidate solution from all existing solutions. The solution vector is analogous to the harmony in music while the local and global search schemes are analogous to musician's improvisations. In comparison to other metaheuristics methods in the literature, HSA imposes fewer mathematical requirements as it can be easily adapted for solving several sorts of engineering optimization challenges [26]. Furthermore, numerical comparisons have demonstrated that the convergence for the HSA is faster than GA [27] which attracts further attention. It has been successfully applied to solve a wide range of practical optimization problems such as discrete and continuous structural optimization [28], parameter estimation of the nonlinear Muskingum model [29], design optimization of water distribution networks [30], vehicle routing [31], combined heat and power economic dispatch [32], design of steel frames [33], or image processing [34] among other. Although the standard HSA presents good search characteristics, several modifications to the original HSA have been proposed in the literature in order to enhance its own features [35].

Similarly, the Artificial Bee Colony (ABC) [36] is another important technique in the field of metaheuristic algorithms. The ABC is inspired by the intelligent behavior of honey bees. ABC consists of three essential components: food source positions, nectar amount, and several honey-bee classes. Each food source position represents a feasible solution for the problem under consideration. The nectar amount for a food source represents the quality of such a solution (given by a fitness value). Each bee class symbolizes one particular operation for generating new candidate food source positions (i.e., candidate solutions). The ABC algorithm starts by producing a randomly distributed

initial population (food source locations). After initialization, an objective function evaluates whether such candidates represent an acceptable solution (nectar amount) or not. Guided by the values of such an objective function, candidate solutions are evolved through different ABC operations (honey-bee types) until a termination criterion is met. The performance of the ABC has been compared to other evolutionary methods such as GA and PSO [37, 38]. The results have shown that ABC produces optimal solutions when it faces multi-modal and noisy optimization problems. Such characteristics have motivated the use of ABC to solve different types of engineering problems within several fields [39-44]. One relevant advantage of the ABC method is that it does not follow a local strategy for computing new solutions. Instead, the ABC method uses a set of operators to build solutions from random operations avoiding falling into local optimal.

Although metaheuristic algorithms are effective alternatives to solve complex optimization problems, there exists the tendency of hybridization of various methods in order to improve the performance and convergence velocity. The opposition-based learning (OBL), that has been initially proposed in [45], is a machine intelligence strategy which considers the current estimate and its correspondent opposite value (i.e., guess and opposite guess) at the same time to achieve a fast approximation for a current candidate solution. It has been mathematically proved [46-48] that an opposite candidate solution holds a higher probability for approaching the global optimum solution than a given random candidate, yet quicker. Recently, the concept of opposition has been used to accelerate metaheuristic-based algorithms such as GA [49], DE [50], PSO [51] and Gravitational Search algorithm (GSA) [52, 53].

On the other hand, digital signal processing is an important tool in electronic and informatics engineering. It permits to extract information from physical systems and transform from analogue to digital signals. Given a digital signal it allows to be manipulated and analyzed by algorithms in a computer. Such algorithms depend on the implementation, but there exist a key step in digital signal processing, it is filtering, that is a process used to remove noise and undesired frequencies that analogue signals may have from the acquisition process [54]. It is important to mention that digital signals can be monodimensional or multidimensional, depending on the number of sources [54]. However the digital filters are used for both types.

Digital filters are classified in two main categories [54]: Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). The output from a FIR filter is computed using

the actual and previous values of input, this fact implies that when doesn't exist any input, the output tends to disappear. On the other hand, the output of an IIR filter is dominated not only by the actual and previous value of the input, but also by previous values of its own output. Such feedback produces that when the input is not present, past outputs continue producing exists in the filter. IIR filter modelling and its identification are important tasks in several fields of engineering, such as digital signal processing, automatic control, communication systems, image processing and noise removal [55]. Identification process consists in to alter the filter coefficients in order to estimate the parameters of an unknown system from the knowledge of the values of its input and output. In order to perform this task it is necessary to minimize an objective function, the Mean Square Error (MSE) that exists between the system response and the output of a modeled filter is typically used in the minimization process.

In order to identify the digital filter coefficients several algorithms using different metaheuristic techniques have been developed. Some examples include the use of Learning Automata (LA) [55], Artificial Immune Algorithm (AIA) [56], GSA [52], PSO [57]. Despite the great success of such approaches, there are some deficiencies that significantly affect their performance [58].

Digital images are the most common multidimensional signals, for instance, a grayscale image is a bi-dimensional signal in which the light intensity levels are quantified. Commonly, an image is represented as  $I(x,y)$ , where the intensity value is obtained by measuring the level at coordinates  $(x,y)$ . The simplest way of representing an image is by a matrix, each element of such matrix corresponds to an intensity value. From this point of view, the digital signal processing techniques are applied to images, however by their own nature, the scientific community has created a new topic called digital image processing. The methods for digital image processing play an important role in computer sciences because it is possible to take several samples from a specific environment. The intention of having different images is to analyze them and to extract important features from the objects that contain, similar to the human eye. An example of this is the tridimensional surface reconstruction or terrains in robotics [59]. In this way, the digital image processing systems are very important in engineering processes. In image analysis there exist different phases, they permit the image manipulation and the object analysis in each scene. Such phases range from the classification of intensity levels (segmentation) until identification and object detection. Digital image processing systems must be able for adapting to the

conditions of each image; here it is possible to integrate them with intelligent algorithms. A key issue to allow the implementation of metaheuristic algorithm in image processing is the good definition of an objective function.

Segmentation is one of the most important tasks in image processing that endeavors to identify whether the intensity of a pixel corresponds to a predefined class. Thresholding is the easiest method for segmentation, as it works taking a threshold ( $th$ ) value so that pixels whose intensity value is higher than  $th$  are labeled as a first class while the rest corresponds to a second class label. In recent years, image processing has been applied to different areas such as engineering, medicine, agriculture, and so forth. Since most of such implementations use a thresholding (TH) methodology, several techniques have been studied. Generally, TH methods are divided into parametric and nonparametric [60–63]. Parametric approaches need to estimate values of a probability density function to model each class. The estimation process is time consuming and computationally expensive. On the other hand, the TH nonparametric employs several criteria such as the between-class variance, the entropy, and the error rate [64–6] in order to verify the quality of a  $th$  value. These metrics could also be used as optimization functions since they result as an attractive option due to their robustness and accuracy. There exist several classical thresholding methods [67]. Two of them are classic references. The first, proposed by Otsu in [64] that maximizes the variance between classes while the second method, submitted by Kapur et al. in [65], uses the maximization of the entropy to measure the homogeneity among classes. Several metaheuristic algorithms have been proposed for image segmentation, for instance PSO and ABC that are used to find the best thresholding values [68]. Both methods are used to find the optimal multilevel threshold points by using the Kapur's entropy as a fitness function. In [70], the optimal segmentation threshold values are determined by using the bacterial foraging algorithm (BFA) [69]. Such method aims to maximize the Kapur's and Otsu's objective functions by considering a set of operators that are based on the social foraging behavior of the bacteria *Escherichia Colli*.

Object detection in images, allows providing important information about the analyzed environment. It is possible to say that object detection is one of the main objectives in digital image processing, due this task it is the most similar to the human eye. It is applied in different problems, such as surveillance, visual inspection, autonomous vehicle navigation, medical image analysis, etc.

Metaheuristic algorithms have proved to be useful in several application areas [71] with relevant contributions to optimization, pattern recognition, shape detection, and machine learning. In particular, it has gained considerable research interest from the computer vision community as nature-based algorithms have successfully contributed to solve challenging computer vision problems, for example their application in the detection of image changes [72].

On the other hand, white blood cells, also known as leukocytes, play a significant role in the diagnosis of different diseases. Although digital image processing techniques have successfully contributed to generate new methods for cell analysis, which, in turn, have lead into more accurate and reliable systems for disease diagnosis, however, high variability on cell shape, size, edge, and localization complicates the data extraction process. Moreover, the contrast between cell boundaries and the image's background may vary due to unstable lighting conditions during the image acquisition process. Many works have been conducted in the area of blood cell detection. In [73], a method based on boundary support vectors is proposed to identify white blood cells. In such approach, the intensity of each pixel is used to construct feature vectors, whereas a support vector machine (SVM) is used for classification and segmentation. By using a different approach, Wu et al. [74], developed an iterative variant from the Otsu method based on the circular histogram for leukocyte segmentation. According to such technique, the smear images are processed in the Hue-Saturation-Intensity (HSI) space by considering that the Hue component contains most of the white blood cells information. One of the latest advances in white blood cell detection research is the algorithm proposed by Wang et al. [75] which is based on the fuzzy cellular neural network (FCNN). Although such method has proved successful in detecting only one leukocyte in the image, it has not been tested over images containing several white cells. Moreover, its performance commonly decays when the iteration number is not properly defined, yielding a challenging problem itself with no clear clues on how to make the best choice.

Since blood cells can be approximated with a quasi-circular form, a circular detector algorithm may be handy. The problem of detecting circular features holds paramount importance for image analysis, in particular for medical image analysis [76]. The circle detection in digital images is commonly performed by the circular Hough transform [77]. A typical Hough-based approach employs an edge detector whose information guides the inference for circle locations and radius values. Peak detection is



then performed by averaging, filtering, and computing the histogram of the transformed space. As an alternative to Hough transform-based techniques, the circle detection problem has also been handled through optimization methods. In general, they have demonstrated to deliver better results than those based on HT considering accuracy, speed, and robustness [78]. Such approaches have produced several robust circle detectors using different optimization algorithms such as GA [78], HSA [79], DE [80] and EMO [81].

As previously mentioned, object detection and recognition in digital images are important fields of research in computer vision and image processing. Template matching (TM) is an image processing technique that aims to find objects in images by defining the best-possible resemblance between a subimage, known as the template, and its coincident region within a source image. In general, TM involves two critical points, namely: the similarity measurement and the search strategy [82]. Although several metrics are known to evaluate the similarity between two images, the most important are: the sum of absolute differences (SAD), the sum of squared differences (SSD), and the normalized cross-correlation (NCC). The calculation of such metrics is computationally expensive and represents the most time-consuming operation in the TM process [83]. Although NCC, SAD, and SSD allow adequate assessment of the similarity between two images, the NCC is broadly used due to its robustness. On the other hand, the full search algorithm is the simplest search strategy that can deliver the optimal detection with respect to a maximal NCC coefficient as it checks all pixel-candidates one by one. Unfortunately, such exhaustive search and the NCC computation, at each checkpoint, the TM method becomes extremely computational expensive and seriously constrains its use in image-processing applications. Recently, several TM algorithms, based on evolutionary approaches, have been proposed to reduce the number of NCC operations (objective function evaluations) by calculating only a subset of search locations. Such approaches have produced several robust detectors that employ different optimization methods such as genetic algorithms (GA) [84], the particle swarm optimization (PSO) [85, 86], and the imperialist competitive algorithm (ICA) [87]. Although such algorithms allow a reduction in the number of search locations, they do not effectively explore the whole region and often suffer premature convergence, which conducts to suboptimal detections.

Continuing with the metaheuristic algorithms applications in engineering topics, there exist several areas which by nature are prioritized attracting the attention of the scientific community. The goal of research in such topics is to solve the challenges that

contribute to the benefit of humanity. An example of this is the energetic crisis increased in recent years and directly affecting the prices of the traditional energy sources (fossil based). Even though, the use of these sources arises substantially the pollution level all over the world. The search of alternative and renewable energies is mandatory in order to preserve the environment and recover the economy of the world. Solar energy has attracted the attention of several researchers due that it exists every day during the whole year. The main application of this kind of energy is to provide electrical energy that can be used in domestic and industrial applications. The first use of solar energy was in space systems (vehicles and satellites) in the late 1950's. It was chosen because doesn't require maintenance for long periods [88, 89]. During the years photovoltaic cells have been used to power generation from solar energy, its use increasing substantially in the last 10 years [89-90]. The design of photovoltaic cells consists in two steps: the mathematical model formulation and the parameter estimation of the solar cells. Several approaches have been proposed in order to represent the behavior of solar cells under different working conditions [91-97]. However, in practical terms there exist two equivalent electronic circuits that model the behavior of solar cells. Such circuits are known as: single diode and double diode models [98]. No matter the model selected to estimate or identify some parameters, such as photo-generated current, diode saturation current, series resistance and diode ideality factor. The methods employed to solve the problem of parameter identification of solar cell using diode models can be divided in two groups: deterministic (analytical) and metaheuristic methods. In the deterministic group some approaches have been proposed, such as a modified least error squares estimation technique, based on Newton's theory to obtain the solar cells parameter [99]. Another interesting approach called Co-content function is based on Lambert function to extract the parameters of solar cells [100]. A comparative study of three methods for parameter identification is presented in [14], including techniques as: curve-fitting, iterative and analytical five points methods. On the other hand, in the last years the increase of the computational power made possible the use of alternative optimization methods as metaheuristic algorithms. For instance, the use of the classical GA to estimate the parameters of diode models [102]. AlRashid et, al. [89] proposed a GA with fixed pattern search to improve GA capabilities to find the best values of solar cells models. Another method is the PSO that has been used for solving this problem [97], in the same way a modification of PSO is proposed by Wei et, al. [103].

## 1.2 Problem statement

The methods outlined in section 1.1 have been extensively tested and the results have shown sufficient capabilities to find solutions to the proposed applications. However, after an extensive analysis, it is possible to say that they have deficiencies, which have been addressed in this work proposing several solutions. This research is divided into four parts, in which the efficiency and usefulness of metaheuristic techniques are proved, specifically in the areas of signal processing and digital images. Moreover their applicability in selected engineering problems is also verified. Thus, the problems addressed regarding the mentioned topics are described as follows:

*Metaheuristic algorithms:* there exists sufficient evidence to corroborate the efficiency of metaheuristic optimization algorithms. Regardless of the metaphor on which they are based, these methods must have a balance between exploration and exploitation processes. However, depending on the problem to be solved, some metaheuristic techniques are more focused on exploration than exploitation, increasing the computational cost. An example of this is the EMO algorithm, which despite its strong performance has some shortcomings. EMO algorithm is based on four main phases: initialization, local search, calculation and movement. The local search procedure is a stochastic search in several directions over all coordinates of each particle. EMO's main drawback is its computational complexity resulting from the large number of iterations which are generally required during the searching process. The issue becomes worst as the dimension of the optimization problem increases. Several approaches, which simplify the local search, have been proposed in the literature to reduce EMO's computational effort. Guan et al. [104] proposed a discrete encoding for the particle set in order to reduce search directions at each dimension. Other proposed method includes a new local search approach which is based on a fixed search pattern and a shrinking strategy that aims to reduce the population size as the iterative process progresses [105, 106]. Additionally, a modified local search phase that employs the gradient descent method is adopted to enhance its computational complexity [14]. Although all these approaches have improved the computational time, which is required by the original EMO algorithm, recent works [23,107] have demonstrated that reducing or simplifying EMO's local search processes also affects other important properties, such as convergence, exploration, population diversity and accuracy. Based on these facts, the problem to be treated in metaheuristic algorithms is to reduce the high

computational cost of the original EMO that is generated in the local search phase. The proposed solution should not affect the efficiency of the original EMO operators.

*Digital filter design:* regardless of which type of filter is selected (FIR or IIR) it is necessary to modify its coefficients to obtain a desired output. In order to verify the filter quality, using a determined set of coefficients, the MSE is used. In this research IIR filters are the only ones considered. Nevertheless, taking into account the wide range of applications of the IIR filters, identification and modeling is considered a complex process requiring global optimization algorithms able to operate on multimodal error surfaces typical of these processes. In section 1.1 are mentioned some implementation in filters using metaheuristic algorithms. However, the work of Shilane et al. [108] demonstrates that some deficiencies significantly affect their performance. One important deficiency is the big amount of iterations and population elements employed to find a solution close to the global optimal, affecting directly the computational cost of the algorithm [109]. In addition, for most of these methods, convergence is not sufficiently proven, therefore there is no guarantee that the values found as a solution, can be considered the best. In this sense the problem in the filter design is clearly defined: to find and implement an optimization algorithm which is able to model high precision IIR filters, with no impact in a high computational cost.

*Digital image processing:* as mentioned before, segmentation is one of the most important tasks in image processing, for that reason it is highly studied in the literature. In this work the non-parametric approaches are analyzed, specifically the popular metrics proposed by Otsu [64] and Kapur [65]. As mentioned in section 1.1, these metrics could be used as objective function. In this case, the segmentation is an iterative process and its goal is to find the values that maximize the selected function. The efficiency and accuracy of both Otsu's and Kapur's methods have been already proved for a bi-level segmentation [62]. Although they can be expanded for multilevel segmentation, their computational complexity increases exponentially when a new threshold is incorporated [63]. Although the use of several metaheuristic algorithms has been proposed to find multiple thresholds, the results lack of accuracy, robustness and computational cost continues being high. An example of this is the analysis proposed by Hammouche et al. [110] that provides an interesting survey of how different metaheuristic algorithms are used to solve the Kaptur's and Otsu's problems. This survey provides evidence of the efficiency of metaheuristics in segmentation, however it shows their deficiencies. In this sense, the problems to be solved

within image segmentation are: find an efficient method to optimize the objective function proposed by Otsu and Kapur. Such technique must be competitive, robust enough and offer accurate results without increasing the computational cost.

On the other hand, in section 1.1, under the topic of medical imaging, and in particular for the detection of white blood cells, it was mentioned that it is possible to perform this task by using the circular Hough transform. However, such approach requires a large storage space given by the required 3D cells to cover all parameters ( $x$ ,  $y$ ,  $r$ ). It also implies a high computational complexity yielding a low processing speed. The accuracy of the extracted parameters for the detected circle is poor, particularly in presence of noise [111]. For a digital image holding a significant width and height and a densely populated edge pixel map, the required processing time for circular Hough transform makes it prohibitive in real time applications. In order to overcome such a problem, some other researchers have proposed new approaches based on the Hough transform, for instance the probabilistic Hough transform [112, 113], the randomized Hough transform [114] and the fuzzy Hough transform [115]. Alternative transformations have also been presented in the literature, as the one proposed by Becker et al. in [116]. Although those new approaches demonstrated better processing speeds, as compared to the original Hough transform, they are still very sensitive to noise. Although detection algorithms based on the optimization approaches present several advantages as compared to those based on the Hough transform, they have been scarcely applied to blood cells detection. One exception is the work presented by Karkavitsas and Rangoussi [76] that solves the blood cells detection problem through the use of GA. However, since the evaluation function, which assesses the quality of each solution, considers the number of pixels contained inside a circle with fixed radius, the method is prone to produce misdetections, particularly for images that contained overlapped or irregular blood cells. In this way, the problems related with blood cells detection should accomplish some requirements: be robust, able to detect or interact with cells imbedded in complex images and the computation time must be less than the proposals based on the Hough transform

In the task of object localization and recognition in digital images, template matching and the normalized cross-correlation (NCC) are used. However, because of exhaustive search, it excessively consumes computational resources; metaheuristic tools considering the NCC as objective function are used as an alternative. Thus, the main goal is the use of metaheuristic techniques to reduce the number of NCC operations by calculating

only a subset of search locations. However, a particular difficulty in applying a metaheuristic algorithm to discrete optimization problems, such as TM, is the multiple evaluation of the same individual. In discrete optimization problems, the search space is composed of a set of finite solutions, which implies the use of random numbers for the calculation of new individuals. So, they may encounter the same solutions (repetition) that have been visited by other individuals at previous iterations, particularly when individuals are confined to a small area [117]. Evidently, such a fact seriously constrains its performance considering that fitness evaluation is computationally expensive to calculate. In this way, in object localization and recognition, the main problem is to find and to implement a search method that reduces the number of NCC evaluations and operates efficiently in discrete optimization problems. This method must offer a good relationship between computation resources and results accuracy..

*Alternative energies:* in solar energy, the topic is the parameter estimation in photovoltaic cell models. Diode models have two different sets of parameters that must be identified: five for the single diode and seven for the double diode. The main problem is to identify the optimal parameter values which, when applied to the selected model, produce the best possible approximation to the experimental data obtained by the true solar cell [98]. As was mentioned in section 1.1, traditional methods have been implemented for solar cells design. However, they require that the problem has some mathematical features such as, continuity, differentiability and convexity. That means, it is necessary to assume some previous considerations before to the initialization of the estimation process. This represents a complex problem, requiring of a human expert to make the adjustment. Besides the computational cost involved on its implementation, they are highly susceptible to the initialization process, therefore it is possible to say that the success or failure depends on this step [98, 118]. The obtained results have a leak of accuracy and cannot be repeated. The major problem of parameter estimation in solar cells is the non-linearity of the objective function, for that reason it results difficult the use of traditional mechanisms [118]. On the other hand, metaheuristic algorithms have been proposed for this problem. Such approaches provide alternatives for the use of deterministic methods because they don't need any previous configuration. However, the non-linearity of the problem makes complicated to find an accurate solution at each iteration. In summary, for the field of solar energy, the problem is to find a metaheuristic algorithm able to accurately estimate the values of the solar cells, ensuring that the results will be repeated every time it runs and should also be adaptable to changes produced in the diode model to be estimated.

### 1.3 *Research objectives*

Once the background has been analyzed and the problem identified, for each topic, this research raises two types of objectives: general and specific.

#### 1.3.1 *General objectives*

1. Adapt metaheuristic algorithms to improve their performance.
2. Develop methods capable of solving signal and digital image processing problems, using metaheuristic algorithms.

#### 1.3.2 *Specific objectives*

The following is a list of approaches to:

1. Modify the standard EMO algorithm using opposition-based learning strategy, in order to reduce the computational cost required in local search without affecting the capabilities of the original EMO algorithm [119].
2. Present a new proposal for the identification and modeling of digital IIR filters based on the EMO algorithm [120].
3. Propose a new multilevel segmentation algorithm for digital images, which combines the features of the original HSA with the methodologies proposed by Otsu and Kapur [121].
4. Implement the EMO algorithm as circles detector for identification of blood cells in medical images [122].
5. Propose a new objective function that improves the accuracy and robustness in the detection of blood cells, the function is used in the detector proposed in the specific objective 3 [123].
6. Develop a method for locating and recognizing objects in digital images using a template matching technique and a modified EMO algorithm [124].
7. Create an algorithm for the design of solar cells by applying the ABC method to estimate parameters of the model based on diodes [125].
8. Perform relevant experiments to objectives 1-7.
9. Analyze the experimental results obtained in the specific objective 8 and compare them with some other existing methods in the literature.
10. Divulge and disseminate research results through scientific publications or other media [119-126].

#### 1.4 Research contributions

In the course of this research it has been possible to provide solutions to the proposed problems. After analyzing the experimental results and with the aim of presenting alternatives to the scientific community to guide for future work, the following journal publications have been produced:

- 1) Cuevas, E., Oliva, D., Zaldívar, D., Pérez-Cisneros, M. Pajares, G. (2012). Opposition-Based Electromagnetism-Like for Global Optimization. *Int. J. of Innovative Computing Information and Control*, 8(12), 8181-8198.

This paper presents an enhanced EMO algorithm called OBEMO, which employs the Opposition-Based Learning (OBL) approach to accelerate the global convergence speed. The proposed OBEMO method significantly reduces the required computational effort yet avoiding any detriment to the good search capabilities of the original EMO algorithm. Experiments are showing that OBEMO obtains promising performance for most of the discussed test problems. This method is described in chapter three.

- 2) Cuevas, E., Oliva, D., Díaz, M., Zaldívar, D., Pérez-Cisneros, M., Pajares, G. (2013). White Blood Cell Segmentation by Circle Detection Using Electromagnetism-Like Optimization. *Computational and Mathematical Methods in Medicine*. 395071, 2013.

This paper presents an algorithm for the automatic detection of white blood cells embedded into complicated and cluttered smear images that considers the complete process as a circle detection problem. The approach is based on the EMO algorithm and uses an objective function which measures the resemblance of a candidate circle to an actual blood cell. Experimental results from blood cell images with a varying range of complexity are included to validate the efficiency of the proposed technique regarding detection, robustness, and stability. This technique is described in chapter three.

- 3) Oliva, D., Cuevas, E., Pajares, G., Zaldívar, D. (2014). Template Matching using an improved Electromagnetism-Like algorithm. *Journal of Applied Intelligence*. 41 (3), 791-807.

This paper introduces a local search modification in the EMO algorithm for object detection in images using template matching. This method uses as objective function the normal cross correlation (NCC) that exists between a template and a section of the image. This procedure is presented in chapter three.



- 4) Oliva, D., Cuevas, E., Pajares, G., Zaldívar, D., Pérez-Cisneros, M. (2013). Multilevel thresholding segmentation based on harmony search optimization. *Journal of Applied Mathematics*. 575414, 2013.

In this paper a multilevel segmentation algorithm (MT) based on Harmony Search (HSA) is presented. This approach combines the HSA search skills, with the objective functions proposed in popular segmentation methods Otsu and Kapur. The proposed algorithm is presented in chapter four.

- 5) Oliva, D., Cuevas, E., Pajares, G. (2014). Parameter identification of solar cells using artificial bee colony optimization. *Energy Journal*. 72 (1), 75-82.

In this paper it is proposed the implementation of the Artificial Bee Colony (ABC) for the identification and modelling of solar cells. The model of solar cells used is based on diodes. The task ABC is to find the best parameter that give an efficient output response of voltage-current. The root mean square error (RMSE) is used as objective function. The implementation of this method is detailed in chapter five.

- 6) Oliva, D., Cuevas, E., Pajares, G., Zaldívar, D., Osuna, V. (2014). A Multilevel Thresholding algorithm using electromagnetism optimization. *Neurocomputing*. 139, 357-381.

In this paper a multilevel segmentation algorithm (MT) based on EMO is presented. This method combines the search skills of EMO, with the objective functions proposed by the popular segmentation methods Otsu and Kapur. The implementation of this work is similar to reference [121].

- 7) Cuevas, E., Zaldívar, D., Pérez-Cisneros, M., Oliva, D. (2013). Block-matching algorithm based on differential evolution for motion estimation. *Engineering applications of artificial intelligence*. 26, 488-498.

In this paper an algorithm based on differential evolution (DE) and the block matching technique (BM) is proposed for motion estimation. The algorithm estimates the values for some locations of the sum of absolute differences (SAD), which is also considered objective function. In order to estimate the values, there are used the previous SAD values computed for the near positions. This method is not described in the thesis however some ideas are taken to develop the work presented in [123].

- 8) Cuevas-Jiménez, E., Oliva-Navarro, D. A. (2013). Identificación y modelado de Filtros IIR usando un algoritmo inspirado en el Electromagnetismo. *Ingeniería, Investigación y Tecnología*. 14(1), 125-138.

In this paper a method based on EMO for identification and modeling of infinite impulse response filters is proposed. The simulation results show that the proposed approach has better accuracy and convergence than those obtained by methods reported in the literature to perform the same task. The description of the proposed approach is detailed in chapter three.

## 2 Methods and results

In this section the proposed implementations and modification of the optimization algorithms are briefly described with the results produced. Firstly the modification of EMO, using the OBL strategy is introduced, then the implementation of EMO for digital filter design is presented, after that the use of EMO in image processing is described for the problem of blood cells detection and automatic object detection. Moreover, for image segmentation the use of HSA is described and finally the application of ABC for solar cells modelling is described. In order to show the performance of the proposed method for each problem, evidences about the experimental results are presented.

### 2.1. *Opposition-based electromagnetism-like optimization (OBEMO)*

Metaheuristic methods start by considering some initial solutions (initial population) and trying to improve them toward some optimal solution(s). The process of searching ends when some predefined criteria are satisfied. In the absence of a priori information about the solution, random guesses are usually considered. The computation time, among others algorithm characteristics, is related to the distance of these initial guesses taken from the optimal solution. The chance of starting with a closer (fitter) solution can be enhanced by simultaneously checking the opposite solution. By doing so, the fitter one (guess or opposite guess) can be chosen as an initial solution following the fact that, according to probability theory, 50% of the time a guess is further from the solution than its opposite guess. Therefore, starting with the closer of the two guesses (as judged by their fitness values) has the potential to accelerate convergence. The same approach can be applied not only to initial solutions but also to each solution in the current population.

Similarly to all metaheuristic-based optimization algorithms, two steps are fundamental for the EMO algorithm: the population initialization and the production of new generations by evolutionary operators. In the approach, the OBL scheme is incorporated to enhance both steps. However, the original EMO is considered as the main algorithm while the opposition procedures are embedded into EMO aiming to accelerate its convergence speed. Figure 3.4 shows a data flow for the EMO and the OBEMO algorithm, which allows a comparison between both methods.

In order to test the algorithm's performance, the proposed OBEMO is compared with the standard EMO and other existing EMO-based algorithms. All algorithms face 14 benchmark problems. The algorithms are: 1) standard EMO algorithm [13]; 2) hybridizing EMO with descent search (HEMO) [14]; 3) EMO with fixed search pattern (FEMO) [29]; 4) the proposed approach OBEMO [119].

The results, shown in Table 3.3, are evaluated assuming the averaged best value  $f(x)$  and the averaged number of executed iterations. Figure 3.5 shows the optimization process for functions  $f_3$  and  $f_6$ . Such function values correspond to the best case for each approach that is obtained after 35 executions.

## 2.2. IIR filter modeling using electromagnetism-like optimization algorithm

The process of filter modeling and identification consists in the modification of the filter coefficients until they are closer to an unknown system. First, it is required to define the representation of the particles under the EMO approach. Particles are randomly generated inside a bounded space. For IIR filter modeling, the number of dimension depends on the number of elements in the numerator and denominator of the filter. The particles are stored in a population vector; the quality of each solution is evaluated through the objective function. It measures the fitness of the response between an unknown system and the estimated filter. Each particle is tested in the IIR filter and produces an output signal that must be compared with the outputs of an unknown system. The mean square error (MSE) is used to compute the outputs difference, in other words, it is the objective function. Once all particles are evaluated, the element with the minimum MSE value is considered as the best particle. All particles are modified by EMO operators at each iteration; the aim of such modification is to find the best solutions according the MSE. The process is repeated until a stop criterion is met; it could be a predefined number of iterations.

The proposed method for IIR filter modeling is tested using a set of benchmark problems. Moreover it is compared with a similar implementation based on PSO. The test problems are: 1) modeling a second order plant using a first order model; 2) modeling a second order plant using a second order model; 3) modeling a third order plant using a second order model; 4) modeling a higher order plant using a higher order model.

An example for the first problem is expressed by equations (3.12) and (3.13). Equation (3.12) defines the transfer function for the plant to be modeled. Meanwhile equation (3.13) presents the transfer function of the filter that should model the plant. In Table 3.9 are presented the final values for the model parameter obtained by EMO, such values are compared with the real values presented by the benchmark problem.

Figure 3.8(a) shows the evolution of MSE (objective function) along 3000 iterations, on the other hand the evolution of the parameters  $a$  and  $b$  are shown in Figure 3.8(b). In both figures it is possible to infer that the values achieve the stabilization in the first 500 iterations and after that values are only adjusted for convergence. In both cases the designs are for a second order plant modeled by a first order model.

### 2.3. *White blood cell detection using electromagnetism-like optimization*

In order to detect the white blood cells in medical images, an algorithm based on EMO algorithm is proposed. Basically this method is based on two main steps: image preprocessing and cells detection using a circle detector mechanism. The image preprocessing applied to obtain two new images: the segmented image and its corresponding edge map. The goal of the segmentation is to isolate the white blood cells from other structures. Although a simple histogram-based thresholding method can be used to segment this kind of blood cells, in this work, the diffused expectation maximization (DEM) has been used to ensure better results [179]. Once the image is segmented, the edge map is computed using the morphological edge detection procedure [145]. Now, in order to detect the global cells a modified circle detector based on EMO is applied. This approach uses the encoding of three edge points that represent candidate circles in the image. In the original EMO-based circle detector, the quality of each individual is calculated by using an objective function which evaluates the existence of a candidate circle considering only information from the edge map (shape structures). In this research, a new objective function has been derived to measure the resemblance of a candidate circle to an actual white blood cell based on the information from the edge map and the segmented image. Such new objective function takes into consideration not only the information provided by the edge

map but also the relationship among the pixels falling inside the candidate circle which is contained in the segmented image, validating the existence of the blood cell.

To illustrate the functionality of the new objective function, Figure 3.21 presents a detection procedure which considers a complex image. Figure 3.21(a) shows the original smear image containing a white blood cell and a stain produced by the coloring process. Figures 3.21(b) and 3.21(c) represent the segmented image and the edge map, respectively. Since the stain contained in the smear image, Figure 3.21(a) displays similar properties than a white blood cell, it remains as a part of the segmented image. In Figure 3.21(d), the detected circle presents a coincidence of 37 different pixels in the edge map. Figure 3.21(e) presents the detected result that has been produced by the modified EMO-based circle detector. In such figure, the detected circle matches with only 32 pixels of the edge map. However, it is considered as the best instance due to the relationship of its internal pixels (the white pixels are much more than the black pixels). Finally, Figure 3.21(f) shows the final detection results over the original smear image.

#### 2.4. *Template matching using an improved electromagnetism-like optimization algorithm*

In this section, the use of modified EMO method is proposed as a search strategy to solve the task of template matching. The new implementation avoids two traditional drawbacks of the original EMO algorithm: the use of a large number of fitness evaluations and the re-evaluation of already-visited individuals. Two modifications are presented to enhance the EMO performance. Such modifications involve the construction of a new local search procedure and the incorporation of a memory to store the objective function values when calculated. As a result, the new EMO algorithm can substantially reduce the number of fitness function (NCC under the TM approach) evaluations while preserving the good search capabilities of the original EMO.

The proposed local search approach is used to exploit prominent solutions. Such a local search mechanism reduces the computational effort produced by the traditional algorithm proposed in [13]. The new procedure sorts the population according to their fitness values. Only best elements of the population are used; however, they must be enough separated from each other to be selected. A predefined distance is used that determines the magnitude of separation between the particles. However, such distance pursues a double purpose. First, it is used to determine which particles are taken from the sorted population. The selected elements must be separated by at least twice the

exploitation distance value. The second purpose is to create the subspaces from which new particles will be taken. Therefore, inside each subspace, a specific number of random samples are selected. From all samples, the best one, according to its objective function value, is selected. To update the positions of the EMO particles, the best samples from each space are compared with their respective original particles from the sorted population. If the sample has a better objective function value than its original particle, its position is updated; otherwise, it does not suffer any modification.

On the other hand, to enhance the performance of the searching strategy, the number of normal cross correlation (NCC) evaluations is reduced by incorporating a fitness memory (FM) to store the NCC (objective function) values previously visited, avoiding the re-evaluation of the same particle positions. The FM memory contains a list that includes the search position and its corresponding NCC value. Therefore, for a given search position, the FM memory is verified in advance to check whether such a position is already contained, indicating that an evaluation is not further required. Otherwise, the NCC of such a search position is calculated and stored at the FM memory for its posterior use.

To verify the feasibility and effectiveness of the proposed algorithm, a set of comparative experiments with other TM algorithms have been conducted. Such experiments have considered a set of benchmark images displayed in Table 3.20 with their respective templates and sizes. Table 3.20 also presents the NCC surface produces after an exhaustive search. The experimental results and comparisons are displayed in Table 3.21 and Figure 3.23, where it is possible to see the performance of the proposed approach in terms of accuracy and robustness.

## 2.5. *Multilevel thresholding using harmony search algorithm (HSMA)*

The proposed approach, called the harmony search multi-thresholding algorithm (HSMA), combines the original harmony search algorithm (HSA) and the Otsu's and Kapur's methodologies. The proposed algorithm takes random samples from a feasible search space inside the image histogram. Such samples build each harmony (candidate solution) in the HSA context, whereas its quality is evaluated considering the objective function that is employed by the Otsu's or the Kapur's method. Guided by these objective values, the set of candidate solutions are evolved using the HSA operators until the optimal solution is found. The approach generates a multilevel segmentation algorithm which can effectively identify the threshold values of a digital image within a reduced number of

iterations. Experimental results over several complex images have validated the efficiency of the proposed technique regarding accuracy, speed and robustness.

To evaluate the stability and consistency, it has been computed the standard deviation (STD) from the results obtained in 35 executions. Since the STD represents a measure about how the data are scattered, the algorithm becomes more instable as the STD value increases [70]. On the other hand, as an index of quality, the peak-to-signal ratio (PSNR) is used to assess the similarity of the segmented image against a reference image (original image) based on the produced mean square error (MSE) [68,192].

Figure 4.1 presents different selected images from a test set and their histograms. The results after applying the HSA with the Otsu function (see section 4.2.1) are presented in five tables: table 4.3 presents the numerical results that include the thresholds, PSNR and STD; tables 4.4, 4.5, 4.6, 4.7, 4.8, show the graphical results: image after segmentation, histogram with thresholds and the evolution of the fitness function. On the other hand, for the Kapur function (see section 4.2.2) the results are presented in tables 4.9, 4.10, 4.11, 4.12, 4.13 and 4.14. For these experiments five thresholds were used. A comparison between Otsu and Kapur was performed using a non-parametric statistical proof; the results are shown in table 4.15. Finally tables 4.16 and 4.17 provide evidence of the performance of the proposed approach that was shared with other state-of-the-art methods, here is possible to see the accuracy obtained using HSA.

## 2.6. *Parameter identification of solar cells using artificial bee colony optimization*

The proposed approach encodes the parameters of the solar cell as a candidate solution. The representation of such a candidate solution (food source) depends on the model selected: double diode or single diode. Therefore, each food source uses seven elements for the double diode formulation and five for the single diode model as decision variables within the optimization-based artificial bee colony (ABC) algorithm. The ABC takes random solutions from a feasible space, such solutions or food sources are modified by the operators of ABC. The quality of the food sources is measured by an objective function; the RMSE is used for parameter identification of solar cells.

In order to prove the performance of the proposed approach, the algorithm has been tested using a commercial silicon solar cell (from the R.T.C. Company of France) under the standard test conditions (STC), with a diameter of 57mm. During the data collection process, it is considered that the solar cell operates under the following operating

conditions: 1 sun ( $1000 \text{ W/m}^2$ ) at  $T = 33^\circ\text{C}$ ; however, in order to test the performance of the ABC method the following additional four temperature values have been included,  $T = 50^\circ\text{C}$ ,  $T = 75^\circ\text{C}$  and  $T = 100^\circ\text{C}$ .

In Figure 5.3 are shown the results after applying the proposed algorithm, based on ABC, to identify the parameters of the two solar cells diode models. The results presented are the graphical representation of the output current and the power of the solar cell and the evolution of the RMSE values along the iterations of ABC. All the experimental results are obtained using different temperatures in order to prove the efficiency of the proposed approach under these changes, affecting directly the current and power values of the cell. On the other hand, the evolution of the optimization process shows that the proposed method allows to find appropriate solar cell models in a reduced number of generations. Comparisons with similar approaches are presented in Figure 5.4 that provides evidence of the performances of the ABC based algorithm.

### 3 Conclusions and future work

This section contains the most relevant conclusions derived from the research developed and reflected in the present work. However, there is still great scope for the methods analyzed and studied problems. Thus, according to the methodology the conclusions are divided in two parts: specific and general. In the specific ones, a closure is discussed about the described topics for each one of the addressed problems and how the solutions were found. Meanwhile in the general conclusion, the experiences obtained throughout the research process are described in a generic form. Finally, future work about metaheuristic algorithms and their potential improvements and implementations are described.

#### 3.1. Specific conclusions

The most relevant aspects, regarding implementations or modifications of metaheuristic algorithms, are discussed according to the results in the previous sections. First, a modification of EMO algorithm using the learning rule OBL is analyzed, that was described in section 3.1.1. As a second point, the conclusions of digital filter modeling, using EMO (section 3.1.2) are included. The topic of white blood cells detection in medical images using EMO is discussed in section 3.1.3. Finally, potential improvements of EMO



are described in section 3.1.4 concerning the object localization by using a template matching.

### 3.1.1 *Opposition-based electromagnetism-like optimization (OBEMO)*

The OBEMO algorithm is proposed, it combines the opposition learning strategy OBL with the optimization technique EMO standard. As mentioned above, OBL is a machine learning technique that considers at the same time the current solution and its opposite value to obtain a fast approximation for a given solution. The EMO standard has been improved using the following OBL steps: in the population initialization and to produce new populations. The improved algorithm substantially reduces the computational effort, without affecting or reducing the features of EMO.

A set of 14 benchmark functions is employed for experimental study. The results are supported by a statistically significant test (Wilcoxon rank test), demonstrating that OBEMO is equally accurate than EMO, but requiring a small number of iterations. Similarly, the proposed method is faster than other EMO-based algorithms, discussed in the state-of-the-art, such as HEMO [14] and FEMO [105], despite the speed OBEMO maintains its accuracy.

### 3.1.2 *IIR filter modeling using electromagnetism-like optimization algorithm*

During the EMO algorithm implementations, specifically in this a new method based on EMO is proposed for parameter identification and IIR filter modeling. In this approach, the adaptation of the coefficients is considered as an optimization problem and the mean square error (MSE) between the responses of the unknown system and estimated by the filter, over a period of time, is used as objective function. Guided by the values of the objective function, the set of coefficients that models the filter is obtained using the EMO algorithm until the best possible configuration (global minima) is found.

The performance of the proposed algorithm was experimentally compared with a similar technique based on PSO [57], using a standard test set [177]. From the experimental results the method based on EMO provides more accurate results in the same number of iteration.

Although the proposed algorithm provides evidence of better performance, the aim of this research is to show that systems, based on physical principles as EMO, can be considered as highly attractive for the identification of IIR filters definition.

### 3.1.3 *White blood cells detector using EMO algorithm*

In order to verify the performance of EMO algorithm, an implementation in the area of image processing is proposed. In this section, is introduced an algorithm for the automatic detection of white blood cells contained in complex images from blood smear. This method considers the detection as the problem of identifying circular shapes. The proposed approach is based on the metaheuristic technique called EMO that applies the electromagnetism principles to solve complex optimization problems. The method encodes three non-collinear edge points as candidate circles, which are tested in the edge map from the image that contains the blood cells. From this research, a new objective function is derived, which is proposed to measure the similarity between a candidate circle and a white blood cell, based on map information obtained after segmenting the image edges. Using reference values of the objective function, the set of encoded candidate circles (charged particles) are evolved using the EMO algorithm until they can match the blood cells contained in the edge map.

The performance and effectiveness of the proposed based EMO-based method, has been compared favorably against other detectors of white blood cells proposed in the literature, namely: the approach based on support vector contour [73], the iterative Otsu's method [74], the Wang algorithm [75] and the detector based on genetic algorithms [76]. These methods were compared by considering a large set of images, with different levels of complexity. Experimental results demonstrate the high performance of the proposed technique in terms of accuracy, robustness and stability.

### 3.1.4 *Object localization and detection using an EMO based template matching algorithm*

In order to conclude the implementation and features of EMO in this section a new method is proposed. It reduces the search locations in the process of object localization and detection using template matching. The algorithm uses an improved version of EMO that includes a modification in the local search process to accelerate the search space exploitation. This modification reduces the amount of perturbation of each particle, only on

a small set of random samples. The new EMO algorithm reduces the number of objective function evaluations preserving the search capabilities of the original EMO.

In the proposed approach, particles represent search position which move through the pixel position of the source image. The coefficient values of the normal cross correlation (NCC) are used as objective function and such values influence the charge exerted between particles of the EMO population. The NCC evaluates the particles quality of coincidence that exists between the template image and the coincident region in the source image for a given pixel position (particle). The number of NCC evaluations is also reduced considering a memory, where such values are stored for the positions previously visited, this fact avoids to reevaluate particles already explored. Guided by the objective function values, the candidate positions are modified by the EMO operators at each iteration, this process is repeated until the best resemblance is found. The proposed approach achieves the best balance over other methods, used for solving the same problem, in terms of accuracy and computational cost.

On the other hand, the performance of the proposed algorithm was compared with other similar methods, considering different images that present a variety of formats and level of complexity. Experimental results demonstrate the outperformance of the proposed technique in terms of accuracy and number of NCC evaluations.

### *3.1.5 Multilevel segmentation of digital images using harmony search*

In this section is presented a multilevel segmentation method based on the original harmony search algorithm (HSA). The approach combines the HSA capabilities and the use of the objective functions proposed in the multilevel segmentation method of Otsu and Kapur. In order to measure the performance of this approach the peak signal-to-noise ratio (PSNR) is used, in addition it is also applied to measure the segmentation quality, considering the coincidences between the segmented and original image. In this research work, an implementation of HSA without modifications is proposed. The goal is to demonstrate the method effectiveness when it is applied on image processing tasks.

The study explores the comparison of two versions of the segmentation algorithm based on HSA. One uses as objective function the Otsu's method, while the other uses the Kapur's criteria. The experimental results show that the Otsu function provides better results than the criterion of Kapur. This conclusion was confirmed statistically, based on the Wilcoxon test.

The proposed approach is compared favorably against similar techniques, which are implemented by different optimization algorithms, such as GA, PSO and BF. The efficiency of the algorithm is evaluated in terms of PSNR and standard deviation. The experimental results provide evidence of outstanding performance, accuracy and convergence of the proposed algorithm compared to their counterparts.

### 3.1.6 *Parameter identification of solar cells using ABC*

This section introduces the use of artificial bee colony algorithm (ABC) for accurate estimation of solar cells parameters. In this approach, the estimation process is considered as an optimization problem. Thereby, the method encodes the solar cells parameters as a candidate solution. An objective function evaluates the coincidence quality existing between a candidate solution and the experimental data. Considering the objective function values, the set of candidate solutions are modified using the ABC operations. The aim of this process is to find values holding a higher proximity to the measurements of voltage and current from real solar cells.

The proposed ABC-based algorithm has been compared against similar techniques proposed in the literature, namely: PSO, GA and BFA. Similarly, the efficiency and robustness of the proposed method were evaluated. Experimental results demonstrate outperformance, accuracy and convergence speed of the proposed approach for solving the problem of solar cells design. Therefore, in comparison with existing similar methods in the literature, the use of ABC algorithm is a good alternative, as it allows to find the best parameters using both solar cell diode models.

### 3.2. *General conclusions*

Throughout the sections gathering the outcome of the investigation, several problems in the areas of signal processing, image processing and alternative energy (engineering) were analyzed. The problems presented were selected for two reasons: a) the importance within the above areas and b) the complexity. Based on this, appropriate solutions were generated using metaheuristic optimization techniques. The aim of the proposed approaches was basically to create effective alternatives providing reliable and robust solutions to the problems. The algorithms used were: electromagnetism-like optimization (EMO), harmony search (HSA) and artificial bee colony (ABC). Which were

selected based on the operators that use and acceptance they have had on the scientific community.

In order to verify the performance of metaheuristic algorithms for the selected problems, different sets of experiments were developed, where their capabilities are tested by varying the complexity. In the experimental tests, the proposed algorithms were compared with similar existing techniques included in the state-of-the-art of the application areas. The experimental results were analyzed using non-parametric statistical tests, it is proved that the values obtained did not occur by chance, and therefore they are different enough to be considered better.

Experimental results and comparisons show that the proposed methods perform favorably according to the requirements of each application (computation time, evaluations, accuracy, etc.). The objective of this research is to generate alternative solutions based on metaheuristics algorithms, which can be competitive enough to solve the problems that arise. Based on the results it is clear they can be successfully used in applications beyond mathematics test functions.

On the other hand, it is worth mentioning that due to the good results obtained, six articles have been published in indexed international journals [119-124], thus fulfilling one of the objectives proposed at the beginning of this investigation.

### 3.3. *Future work*

Up to this point the objectives for this research were met, however, within these lines of research there are still some additional aspects to explore. As part of future work, we have planned the creation of a metaheuristic algorithm, whose operators are efficient and simple, thus generating a method with a competitive balance between exploration and exploitation of the search space.

On the topic of image processing, the goal is to continue the analysis of areas in which it is possible to apply optimization techniques. One that is already known is frequency filtering in digital images, where the goal is to find the optimal parameters required to model the desired filter. The issue of medical image analysis is also provided, although nothing in medicine is fully defined. It aims to create applications that allow analyzing the eyeballs in order to extract features that allow detection of diseases, such as glaucoma.

On the other hand, driven by the growth of alternative energies, this area represents a topic of general interest, it is planned to implement metaheuristics algorithms for modeling hydrogen PEM cells. Similarly, continuing analyzing and exploring problems such as the arrangement of wind turbines in wind farms, which is a scope for optimization algorithms.

Finally, it is important to note that although these areas are currently of interest, it is intended to search within engineering issues, problems in which these techniques can provide effective solutions.



## Bibliografía

- [1] S. Tan, X. Cheng and H. Xu, An efficient global optimization approach for rough set based dimensionality reduction, *International Journal of Innovative Computing, Information and Control*, vol.3, no.3, pp.725-736, 2007.
- [2] A. Borji and M. Hamidi, A new approach to global optimization motivated by parliamentary political competitions, *International Journal of Innovative Computing, Information and Control*, vol.5, no.6, pp.1643-1653, 2009.
- [3] C.-N. Yang, K.-S. Huang, C.-B. Yang and C.-Y. Hsu, Error-tolerant minimum finding with DNA computing, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10 (A), pp.3045-3057, 2009.
- [4] C. Xu, J. Wang and N. Shiba, Multistage portfolio optimization with var as risk measure, *International Journal of Innovative Computing, Information and Control*, vol.3, no.3, pp.709-724, 2007.
- [5] J.-F. Chang, A performance comparison between genetic algorithms and particle swarm optimization applied in constructing equity portfolios, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12 (B), pp.5069-5079, 2009.
- [6] W. Gao and H. Ren, An optimization model based decision support system for distributed energy systems planning, *International Journal of Innovative Computing, Information and Control*, vol.7, no.5(B), pp.2651-2668, 2011.
- [7] Y. Takeuchi, Optimization of linear observations for the stationary kalman filter based on a generalized water filling theorem, *International Journal of Innovative Computing, Information and Control*, vol.4, no.1, pp.211-230, 2008.
- [8] A. H. Borzabadi, M. E. Sadjadi and B. Moshiri, A numerical scheme for approximate optimal control of nonlinear hybrid systems, *International Journal of Innovative Computing, Information and Control*, vol.6, no.6, pp.2715-2724, 2010.
- [9] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
- [10] J. Kennedy and R. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, Piscataway, NJ, pp.1942-1948, 1995.
- [11] K. Price, R. Storn and A. Lampinen, *Differential Evolution a Practical Approach to Global Optimization*, Springer Natural Computing Series, 2005.
- [12] M. Dorigo, V. Maniezzo and A. Coloni, Positive feedback as a search strategy, *Technical Report 91-016*, Politecnico di Milano, Italy, 1991.
- [13] S. I. Birbil and S.-C. Fang, An electromagnetism-like mechanism for global optimization, *Journal of Global Optimization*, vol.25, pp.263-282, 2003.
- [14] A. Rocha and E. Fernandes, Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems, *International Journal of Computer Mathematics*, vol.86, pp.1932-1946, 2009.
- [15] A. Rocha and E. Fernandes, Modified movement force vector in an electromagnetism-like mechanism for global optimization, *Optimization Methods & Software*, vol.24, pp.253-270, 2009.
- [16] C. S. Tsou and C. H. Kao, Multi-objective inventory control using electromagnetism-like metaheuristic, *International Journal of Production Research*, vol.46, pp.3859-3874, 2008.
- [17] P. Wu, W.-H. Yang and N.-C. Wei, An electromagnetism algorithm of neural network analysis an application to textile retail operation, *Journal of the Chinese Institute of Industrial Engineers*, vol.21, no.1, pp.59-67, 2004.
- [18] S. I. Birbil, S. C. Fang and R. L. Sheu, On the convergence of a population-based global optimization algorithm, *Journal of Global Optimization*, vol.30, no.2, pp.301-318, 2004.
- [19] B. Naderi, R. Tavakkoli-Moghaddam and M. Khalili, Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan, *Knowledge-Based Systems*, vol.23, pp.77-85, 2010.
- [20] H.-L. Hung and Y.-F. Huang, Peak to average power ratio reduction of multicarrier transmission systems using electromagnetism-like method, *International Journal of Innovative Computing, Information and Control*, vol.7, no.5(A), pp.2037-2050, 2011.



- [21] A. Yurtkuran and E. Emel, A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems, *Expert Systems with Applications*, vol.37, pp.3427-3433, 2010.
- [22] J.-Y. Jhang and K.-C. Lee, Array pattern optimization using electromagnetism-like algorithm, *AEU – International Journal of Electronics and Communications*, vol.63, pp.491-496, 2009.
- [23] C. H. Lee and F. K. Chang, Fractional-order PID controller optimization via improved electromagnetism-like algorithm, *Expert Systems with Applications*, vol.37, pp.8871-8878, 2010.
- [24] E. Cuevas, D. Oliva, D. Zaldivar, M. Pérez-Cisneros and H. Sossa, Circle detection using electromagnetism optimization, *Information Sciences*, vol.182, no.1, pp.40-55, 2012.
- [25] Z.W. Geem, J. H. Kim, and G. V. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [26] M. Mahdavi, M. Fesanghary, and E. Damangir, An improved harmony search algorithm for solving optimization problems, *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.
- [27] K. S. Lee and Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36-38, pp. 3902–3933, 2005.
- [28] K. S. Lee, Z.W. Geem, S.-H. Lee, and K.-W. Bae, The harmony search heuristic algorithm for discrete structural optimization, *Engineering Optimization*, vol. 37, no. 7, pp. 663–684, 2005.
- [29] J. H. Kim, Z. W. Geem, and E. S. Kim, Parameter estimation of the nonlinear Muskingum model using harmony search, *Journal of the American Water Resources Association*, vol. 37, no. 5, pp. 1131–1138, 2001.
- [30] Z. W. Geem, Optimal cost design of water distribution networks using harmony search, *Engineering Optimization*, vol. 38, no. 3, pp. 259–280, 2006.
- [31] Z.W. Geem, K. S. Lee, and Y. J. Park, Application of harmony search to vehicle routing, *American Journal of Applied Sciences*, vol. 2, pp. 1552–1557, 2005.
- [32] A. Vasebi, M. Fesanghary, and S.M. T. Bathaee, Combined heat and power economic dispatch by harmony search algorithm, *International Journal of Electrical Power and Energy Systems*, vol. 29, no. 10, pp. 713–719, 2007.
- [33] S. O. Degertekin, Optimum design of steel frames using harmony search algorithm, *Structural and Multidisciplinary Optimization*, vol. 36, no. 4, pp. 393–401, 2008.
- [34] E. Cuevas, N. Ortega-Sánchez, D. Zaldivar, and M. Pérez- Cisneros, Circle detection by harmony search optimization, *Journal of Intelligent and Robotic Systems*, pp. 1–18, 2011.
- [35] O.M. Alia and R. Mandava, The variants of the harmony search algorithm: an overview, *Artificial Intelligence Review*, vol. 36, no. 1, pp. 49–68, 2011.
- [36] D. Karaboga. An idea based on honey bee swarm for numerical optimization, *technical report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department (2005).
- [37] D. Karaboga, B. Basturk. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8 (1) (2008) 687–697.
- [38] D. Karaboga, B. Akay. A comparative study of artificial bee colony algorithm. *Appl Math Comput*, 214 (2009) 108–132.
- [39] N. Karaboga. A new design method based on artificial bee colony algorithm for digital IIR filters. *J Franklin Inst*, 346 (2009) 328–348.
- [40] Q-K. Pan, M. Fatih Tasgetiren, P.N. Suganthan, T.J. Chua. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, (2011). doi:10.1016/j.ins.2009.12.025.
- [41] F. Kang, J. Li, Q. Xu. Structural inverse analysis by hybrid simplex artificial bee colony algorithms. *Comput Struct*, 87 (2009) 861–870.
- [42] C. Zhang, D. Ouyang, J. Ning. An artificial bee colony approach for clustering. *Expert Syst Appl*, 37 (2010) 4761–4767.
- [43] D. Karaboga, C. Ozturk. A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Appl Soft Comput*, 11 (2011) 652–657
- [44] S.L. Ho, S. Yang. An artificial bee colony algorithm for inverse problems. *Int J Appl Electromagn Mech*, 31 (2009) 181–192.
- [45] H. R. Tizhoosh, Opposition-based learning: A new scheme for machine intelligence, *Proc. of International Conference on Computational Intelligence for Modeling Control and Automation*, pp.695-701, 2005.

- [46] S. Rahnamayn, H. R. Tizhoosh and M. Salama, A novel population initialization method for accelerating evolutionary algorithms, *Computers and Mathematics with Applications*, vol.53, no.10, pp.1605-1614, 2007.
- [47] S. Rahnamayan, H. R. Tizhoosh and M. Salama, Opposition versus randomness in soft computing techniques, *J. Appl. Soft Comput.*, vol.8, pp.906-918, 2008.
- [48] H. Wang, Z. Wu and S. Rahnamayan, Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems, *Soft Computing*, 2010.
- [49] M. A. Iqbal, N. K. Khan, H. Multaba and A. R. Baig, A novel function optimization approach using opposition based genetic algorithm with gene excitation, *International Journal of Innovative Computing, Information and Control*, vol.7, no.7(B), pp.4263-4276, 2011.
- [50] S. Rahnamayan, H. R. Tizhoosh and M. Salama, Opposition-based differential evolution, *IEEE Transactions on Evolutionary Computing*, vol.12, no.1, pp.64-79, 2008.
- [51] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu and M. Ventresca, Enhancing particle swarm optimization using generalized opposition-based learning, *Information Sciences*, vol.181, pp.4699-4714, 2011.
- [52] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, Filter modeling using gravitational search algorithm, *Engineering Applications of Artificial Intelligence*, vol.24, no.1, pp.117-122, 2011.
- [53] B. Shaw, V. Mukherjee and S. P. Ghoshal, A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems, *Electrical Power and Energy Systems*, vol.35, pp.21-33, 2012.
- [54] Oppenheim A.V., Schaffer R.W., Buck J.R. *Discrete Time Signal Processing*, 2a ed., Prentice-Hall, 1999.
- [55] Howell M.N., Gordon T.J. Continuous Action Reinforcement Learning Automata and their Application to Adaptive Digital Filter Design. *Engineering Applications of Artificial Intelligence*, volumen 14, 2001: 549-561
- [56] Kalinli A., Karaboga N. Artificial Immune Algorithm for IIR Filter Design. *Engineering Applications of Artificial Intelligence*, volume 18, 2005: 919-929.
- [57] Lin Y.L., Chang W.D., Hsieh J.G. A Particle Swarm Optimization Approach to Nonlinear Rational Filter Modeling. *Expert Systems with Applications*, volumen 3, 2008: 1194-1199.
- [58] Shilane D., Martikainen J., Dudoit S., Ovaska S. A General Framework for Statistical Performance Comparison of Evolutionary Computation Algorithms. *Information Sciences*, volume 178, 2008: 2870-2879.
- [59] R. Correal, G. Pajares and J. J. Ruz. Automatic expert system for 3D terrain reconstruction based on stereo vision and histogram matching. *Expert Systems with Applications*, Volume 41 4-2 (2014) 2043-2051
- [60] R. Guo and S. M. Pandit, Automatic threshold selection based on histogram modes and a discriminant criterion, *Machine Vision and Applications*, vol. 10, no. 5-6, pp. 331-338, 1998.
- [61] N. R. Pal and S. K. Pal, A review on image segmentation techniques, *Pattern Recognition*, vol. 26, no. 9, pp. 1277-1294, 1993.
- [62] P. K. Sahoo, S. Soltani, and A. K. C. Wong, A survey of thresholding techniques, *Computer Vision, Graphics and Image Processing*, vol. 41, no. 2, pp. 233-260, 1988.
- [63] W. Snyder, G. Bilbro, A. Logenthiran, and S. Rajala, Optimal thresholding: a new approach, *Pattern Recognition Letters*, vol. 11, no. 12, pp. 803-809, 1990
- [64] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man, Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.
- [65] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, *Computer Vision, Graphics, & Image Processing*, vol. 29, no. 3, pp. 273-285, 1985.
- [66] J. Kittler and J. Illingworth, Minimum error thresholding, *Pattern Recognition*, vol. 19, no. 1, pp. 41-47, 1986.
- [67] M. Sezgin, B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1)(2004), 146-165.
- [68] B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, *Applied Soft Computing*, vol. 13, no. 6, pp. 3066-3091, 2012.
- [69] S. Das, A. Biswas, S. Dasgupta, A. Abraham. Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications. *Studies in Computational Intelligence*,

- Volume 203 (2009) 23-55.
- [70] P.D. Sathya and R. Kayalvizhi, Optimal multilevel thresholding using bacterial foraging algorithm, *Expert Systems with Applications*, vol. 38, no. 12, pp. 15549–15564, 2011.
  - [71] J. Liu and K. C. Tsui, Toward nature-inspired computing, *Communications of the ACM*, vol. 49, no. 10, pp. 59–64, 2006.
  - [72] G. Pajares, J. J. Ruz and J. M. de la Cruz. Image change detection from difference image through deterministic simulated annealing. *Pattern Analysis and Applications*, Volume 12 2 (2009) 137-150.
  - [73] M. Wang and R. Chu, A novel white blood cell detection method based on boundary support vectors, in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 2595–2598, San Antonio, Tex, USA, October 2009.
  - [74] J. Wu, P. Zeng, Y. Zhou, and C. Olivier, A novel color image segmentation method and its application to white blood cell image analysis, in *Proceedings of the 8th International Conference on Signal Processing (ICSP '06)*, November 2006.
  - [75] S. Wang, F. L. Korris, and D. Fu, Applying the improved fuzzy cellular neural network IFCNN to white blood cell detection, *Neurocomputing*, vol. 70, no. 7–9, pp. 1348–1359, 2007.
  - [76] G. Karkavitsas and M. Rangoussi, Object localization in medical images using genetic algorithms, *World Academy of Science, Engineering and Technology*, vol. 2, pp. 499–502, 2007.
  - [77] H. Muammar and M. Nixon, Approaches to extending the Hough transform, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '89)*, vol. 3, pp. 1556–1559, May 1989.
  - [78] V. Ayala-Ramirez, C. H. Garcia-Capulin, A. Perez-Garcia, and R. E. Sanchez-Yanez, Circle detection on images using genetic algorithms, *Pattern Recognition Letters*, vol. 27, no. 6, pp. 652–657, 2006.
  - [79] E. Cuevas, N. Ortega-Sánchez, D. Zaldivar, and M. Pérez-Cisneros, Circle detection by harmony search optimization, *Journal of Intelligent & Robotic Systems*, vol. 66, no. 3, pp. 359–376, 2012.
  - [80] E. Cuevas, D. Zaldivar, M. Pérez-Cisneros, and M. Ramírez-Ortegón, Circle detection using discrete differential evolution optimization, *Pattern Analysis & Applications*, vol. 14, no. 1, pp.93–107, 2011.
  - [81] E. Cuevas, D. Oliva, D. Zaldivar, M. Pérez-Cisneros, and H.Sossa, Circle detection using electromagnetism optimization, *Information Sciences*, vol. 182, no. 1, pp. 40–55, 2012.
  - [82] G. Hadi, L. Mojtaba, S. Y. Hadi. An improved pattern matching technique for lossy/lossless compression of binary printed Farsi and Arabic textual images, *Int. J. Intell. Comput. Cybernet*, Volume 2 1 (2009) 120–147.
  - [83] W. Krattenthaler, K. J. Mayer, M. Zeiler. Point correlation: A reduced-cost template matching technique. In: *Proceedings of the First IEEE International Conference on Image Processing*, (1994) 208–212.
  - [84] N. Dong, C.-H. Wu, W.-H. Ip, Z.-Q. Chen, C.-Y. Chan, K.-L. Yung. An improved species based genetic algorithm and its application in multiple template matching for embroidered pattern inspection, *Expert Systems with Applications*, Volume 38 (2011) 15172–15182.
  - [85] L. Fang, D. Haibin, D. Yimin. A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching, *Optik*, 123, (2012), 1955– 1960.
  - [86] C.-H. Wu, D.-Z. Wang, A. Ip, D.-W. Wang, C.-Y. Chan, H.-F. Wang. A particle swarm optimization approach for components placement inspection on printed circuit boards, *J Intell Manuf* 20, (2009), 535–549.
  - [87] D. Haibin, X. Chunfang, S. Liu, S. Shao. Template matching using chaotic imperialist competitive algorithm, *Pattern Recognition Letters*, 31, (2010), 1868–1875.
  - [88] A.R. Jha. *Solar Cell Technology and Applications*. CRC Press, Boca Raton, 2010
  - [89] M.R. AlRashidi, M.F. AlHajri, K.M. El-Naggar, A.K. Al-Othman. A new estimation approach for determining the I–V characteristics of solar cells. *Solar Energy*, 85 (7) (2011) 1543-1550.
  - [90] Renewables, 2010. *Global Status Report*.  
<[http://www.ren21.net/globalstatusreport/REN21\\_GSR\\_2010\\_full.pdf](http://www.ren21.net/globalstatusreport/REN21_GSR_2010_full.pdf)>.
  - [91] K.M. El-Naggar, M.R. AlRashidi, M.F. AlHajri, A.K. Al-Othman. Simulated Annealing algorithm for photovoltaic parameters identification. *Solar Energy*, 86 (1) (2012) 266-274.
  - [92] L. Han, N. Koide, Y. Chiba, T. Mitate. Modeling of an equivalent circuit for dye-sensitized solar cells. *Applied Physics Letters*, 13 (2004) 2433–2435.

- [93] M.G. Villalva, J.R. Gazoli, E.R. Filho. Comprehensive approach to modeling and simulation of photovoltaic arrays. *IEEE Transactions on Power Electronics*, 24 (5) (2009) 1198–1208.
- [94] T. Huld, R. Gottschalg, H.G. Beyer, M. Topic. Mapping the performance of a PV modules, effects of module type and data averaging. *Solar Energy*, 84 (2010) 324–328.
- [95] W. Xiao, M.G.J Lind, W.G Dunford, A. Capel. Real-time identification of optimal operating points in photovoltaic power systems. *IEEE Transactions on Industrial Electronics*, 53 (4) (2006), 1017–1026.
- [96] M. Chegaar, Z. Ouennough, F. Guechi, H. Langueur. Determination of solar cells parameters under illuminated conditions. *Journal of Electron Devices*, 2 (2003) 17–21.
- [97] M. Ye, X. Wang, Y. Xu. Parameter extraction of solar cells using particle swarm optimization. *Journal of Applied Physics*, 105 (9) (2009) 094502–094508.
- [98] Alireza Askarzadeh, Alireza Rezazadeh. Parameter identification for solar cell models using harmony search-based algorithms, *Solar Energy*, 86 (11) (2012) 3241–3249.
- [99] T. Easwarakhanthan, J. Bottin, I. Bouhouch, C. Boutrit. Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers. *Solar Energy*, (4) 1986 1–12.
- [100] A. Ortiz-Conde, F.J. Garcia Sanchez, J. Muci. New method to extract the model parameters of solar cells from the explicit analytic solutions of their illuminated I–V characteristics. *Solar Energy Materials and Solar Cells*, 90 (3) (2006) 352–361.
- [101] D.S.H. Chan, J.R. Phillips, J.C.H. Phang. A comparative study of extraction methods for solar cell model parameters. *Solid-State Electronics*, 29 (3) (1986) 329–337.
- [102] J.A. Jervase, H. Bourdouce, A. Al-Lawati. Solar cell parameter extraction using genetic algorithms. *Measurement Science and Technology*, 12 (11) (2001) 1922–1925.
- [103] H. Wei, J. Cong, X. Lingyun, S. Deyun. Extracting solar cell model parameters based on chaos particle swarm algorithm. In: *International Conference on Electric Information and Control Engineering (ICEICE)*, (2011) pp. 398–402.
- [104] X. Guan, X. Dai and J. Li, Revised electromagnetism-like mechanism for flow path design of unidirectional AGV systems, *International Journal of Production Research*, vol.49, no.2, pp.401–429, 2011.
- [105] A. M. A. C. Rocha and E. Fernandes, Numerical experiments with a population shrinking strategy within a electromagnetism-like algorithm, *Journal of Mathematics and Computers in Simulation*, vol.1, no.3, pp.238–243, 2007.
- [106] A. M. A. C. Rocha and E. Fernandes, Numerical study of augmented Lagrangian algorithms for constrained global optimization, *Optimization*, vol.60, no.10–11, pp.1359–1378, 2011.
- [107] C.-H. Lee, F.-K. Chang, C.-T. Kuo and H.-H. Chang, A hybrid of electromagnetism-like mechanism and back-propagation algorithms for recurrent neural fuzzy systems design, *International Journal of Systems Science*, vol.43, no.2, pp.231–247, 2012.
- [108] Shilane D., Martikainen J., Dudoit S., Ovaska S. A General Framework for Statistical Performance Comparison of Evolutionary Computation Algorithms. *Information Sciences*, volume 178, 2008: 2870–2879.
- [109] Elbeltafia E., Hegazyb T., Grierson D. Comparison Among Five Evolutionary-Based Optimization Algorithms. *Advanced Engineering Informatics*, volumen 19, 2005: 43–53.
- [110] K. Hammouche, M. Diaf, and P. Siarry, A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem, *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 676–688, 2010.
- [111] T. Atherton and D. Kerbyson, Using phase to represent radius in the coherent circle Hough transform, in *Proceedings of IEE Colloquium on the Hough Transform*, pp. 1–4, May 1993.
- [112] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, Volume. 24 6 (1981) 381–395.
- [113] D. Shaked, O. Yaron, and N. Kiryati. Deriving stopping rules for the probabilistic Hough transform by sequential analysis. *Computer Vision and Image Understanding*, Volume 63 3 (1996) 512–526.
- [114] L. Xu, E. Oja, and P. Kultanen, A new curve detection method: randomized Hough transform (RHT), *Pattern Recognition Letters*, vol. 11, no. 5, pp. 331–338, 1990.
- [115] J. H. Han, L. T. Koczy, and T. Poston, Fuzzy Hough transform, in *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 803–808, April 1993.

- [116] J. M. Becker, S. Grousson, and D. Coltuc, From Hough transform to integral geometry, in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS '02)*, vol. 3, pp. 1444–1446, June 2002.
- [117] E. Cuevas. Block-matching algorithm based on harmony search optimization for motion estimation. *Appl Intell*, 39-1 (2013) 165– 183
- [118] Alireza Askarzadeh, Alireza Rezazadeh. Artificial bee swarm optimization algorithm for parameters identification of solar cell models, *Applied Energy*, 102 (2013) 943-949.
- [119] E. Cuevas, D. Oliva, D. Zaldívar, M. Pérez-Cisneros and G. Pajares. Opposition - based electromagnetism-like for global optimization. *International Journal of Innovative Computing, Information and Control*, Volume 8 12 (2012) 456-511.
- [120] E. Cuevas-Jiménez, Diego A. Oliva-Navarro. Identificación y modelado de filtros IIR usando un algoritmo inspirado en el electromagnetismo. *Ingeniería, Investigación y Tecnología*, Volumen 14 1 (2013) 125-138.
- [121] D. Oliva, E. Cuevas, G. Pajares, D. Zaldívar and M. Pérez-Cisneros. Multilevel thresholding segmentation based on harmony search optimization. *Journal of Applied Mathematics*, Volume 2013, Article ID 575414.
- [122] E. Cuevas, D. Oliva, M. Díaz, D. Zaldívar, M. Pérez-Cisneros and G. Pajares. White blood cell segmentation by circle detection using electromagnetism-like optimization. *Computational and Mathematical Methods in Medicine*, Volume 2013, Article ID 395071.
- [123] D. Oliva, E. Cuevas, G. Pajares and D. Zaldívar. Template Matching using an improved Electromagnetism-Like algorithm. *Applied Intelligence*, In press (2014).
- [124] D. Oliva, E. Cuevas and G. Pajares. Parameter identification of solar cells using artificial bee colony optimization. *Energy Journal*, In press (2014).
- [125] E. Cuevas, D. Zaldívar, M. Pérez-Cisneros and D. Oliva. Block-matching algorithm based on differential evolution for motion estimation. *Engineering Applications of Artificial Intelligence*, Volume 26 1 (2013) 488-498.
- [126] D. Oliva, E. Cuevas, G. Pajares, D. Zaldívar and V. Osuna. A Multilevel Thresholding algorithm using electromagnetism optimization. *Neurocomputing*, Volume 139 2 (2014) 357-381.
- [127] Y. Xin-She, Engineering optimization: an introduction with metaheuristics applications, Wiley 2010.
- [128] <http://en.wikipedia.org/wiki/Evolutionaryalgorithm>
- [129] Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. Optimization by Simulated Annealing, *Science*, 220 (1983) (4598): pp. 671-680.
- [130] F. Glover and C. McMillan. The general employee scheduling problem: an integration of MS and AI, *Computers and Operations Research*, 1986.
- [131] Koza, J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection, *MIT Press* 1992.
- [132] McKinnon, K.I.M. Convergence of the Nelder and Mead simplex method to a nonstationary point, *SIAM J Optimization*, 9 (1999): 148-158.
- [133] Q. Xu, L. Wang, N. Wang, X. Hei, L. Zhao. A review of opposition-based learning from 2005 to 2012, *Engineering Applications of Artificial Intelligence*, 29 (2014) 1-12.
- [134] S. Rahnamayan, G. G. Wang, M. Ventresca. An intuitive distance-based explanation of opposition-based sampling, *Applied Soft Computing*, 12 (2012) 2828-2839.
- [135] H.R. Tizhoosh, M. Ventresca. Studies in Computational Intelligence: Oppositional Concepts in Computational Intelligence. *Springer-Verlag*, Berlin 2008.
- [136] H.R. Tizhoosh. Opposition-based learning: A new scheme for machine intelligence. In: *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, 28–30 November 2005, Vienna, Austria, pp.695–701.
- [137] F.S. Al-Qunaieer, H.R. Tizhoosh, S. Rahnamayan. Opposition based computing a survey. In: *Proceedings of the International Joint Conference on Neural Networks*, 18–23 July 2010, Barcelona, Spain, pp.1–7.
- [138] M. Ergezer, I. Sikder. Survey of oppositional algorithms. *International Conference on Computer and Information Technology*, 22–24 December 2011, Dhaka, Bangladesh, pp.623–628.
- [139] M. Imran, H. Jabeen, M. Ahmad, Q. Abbas, W. Bangyal. 2010 Opposition based PSO and mutation operators. *International Conference on Education Technology and Computer*, 22–4 June

- 2010, Shanghai, China, pp.506–508.
- [140] S. Rahnamayan, H. R. Tizhoosh, M. M. A. Salama. Opposition-based differential evolution. *IEEE Trans. Evol. Comput.* 12(1)(2008), 64–79.
  - [141] J. G. Proakis, D. G. Manolakis. Digital signal processing, principles, algorithms and applications. New Jersey, *Prentice Hall*, 1996.
  - [142] J. G. Proakis, D. G. Manolakis. Tratamiento Digital de Señales. *Prentice Hall*, 1998.
  - [143] E. Cuevas, D. Zaldivar, M. Pérez, Procesamiento digital de imágenes con MatLab y Simulink. México, Alfaomega, 2010.
  - [144] Pajares, G. and de la Cruz, J. M. Visión por Computador: imágenes digitales y aplicaciones, RA-MA, Madrid, 2007.
  - [145] R.C. Gonzalez and R.E. Woods, Digital Image Processing. *Addison Wesley*, Reading, MA, 1992.
  - [146] J. S. Weszka and A. Rosenfeld. Threshold evaluation techniques. *IEEE Trans. Syst. Man Cybern.* SMC-8 (1978), 627–629.
  - [147] M. I. Sezan. A peak detection algorithm and its application to histogram-based image data reduction. *Graph. Models Image Process.* 29 (1985), 47–59.
  - [148] J. C. Olivio. Automatic threshold selection using the wavelet transform. *Graph. Models Image Process.* 56 (1994), 205–218.
  - [149] R. Guo and S. M. Pandit. Automatic threshold selection based on histogram modes and a discriminant criterion. *Mach. Vision Appl.* 10 (1998), 331–338.
  - [150] T. W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Trans. Syst. Man Cybern.* SMC-8 (1978), 630–632.
  - [151] M. K. Yanni and E. Horne. A new approach to dynamic thresholding. *EUSIPCO'94: 9th European Conf. Sig. Process.* 1 (1994), 34–44.
  - [152] A. Beghdadi, A. L. Negrata, and P. V. De Lesegno. Entropic thresholding using a block source model. *Graph. Models Image Process.* 57 (1995), 197–205.
  - [153] J. Sauvola and M. Pietaksinen, Adaptive document image binarization. *Pattern Recogn.* 33 (2000), 225–236.
  - [154] I. Dagher, K.E. Tom. WaterBalloons: a hybrid watershed Balloon Snake segmentation. *Image and Vision Computing*, 26 (7) (2008) 905-912.
  - [155] M. Kass, A. Witkin, D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1 (1987) 321-331.
  - [156] Gilmore, M. S., D. R. Thompson, L. J. Anderson, N. Karamzadeh, L. Mandrake, and R. Castaño. (2011), Superpixel segmentation for analysis of hyperspectral data sets, with application to Compact Reconnaissance Imaging Spectrometer for Mars data. *Moon Mineralogy Mapper data, and Ariadnes Chaos, Mars, J. Geophys. Res.* 2116 (2011), E07001, doi:10.1029/2010JE003763
  - [157] X. Ren and J. Malik, “Learning a classification model for segmentation”, *In ICCV*, 2003.
  - [158] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels. *EPFL Technical Report no. 149300*, June 2010.
  - [159] A.P. Moore, S. Prince, J. Warrell, U. Mohammed, G. Jones. Superpixel lattices. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, vol., no., pp.1,8, 23-28 June 2008, doi: 10.1109/CVPR.2008.4587471
  - [160] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* vol. 34, num. 11, p. 2274 - 2282, May 2012.
  - [161] A. Dawoud and A. Netchaev. Preserving objects in Markov Random Fields region growing image segmentation. *Pattern Anal. Appl.* 15, 2 (May 2012), 155-161. DOI=10.1007/s10044-011-0198-x <http://dx.doi.org/10.1007/s10044-011-0198-x>
  - [162] C. Li, C. Xu, C. Gui, and M. D. Fox. Distance Regularized Level Set Evolution and Its Application to Image Segmentation. *IEEE Transactions On Image Processing.* Vol. 19, No. 12, December 2010, 243.
  - [163] P. V. C. Hough. Method and means for recognizing complex patterns. *U. S. Patent* 3, 069 654, December 18, 1962.
  - [164] M. Wolf, G. Noel, R. Stirn. Investigation of the double exponential in the current-voltage characteristics of silicon solar cells. *IEEE Transaction on Electron Devices* 24 (4) (1977) 419–428.
  - [165] T. Easwarakhanthan, J. Bottin, I. Bouhouch, C. Boutrit. Nonlinear minimization algorithm for

- determining the solar cell parameters with microcomputers. *Solar Energy* (4) 1986 1–12.
- [166] L. Sandrolini, M. Artioli, U. Reggiani. Numerical method for the extraction of photovoltaic module double-diode model parameters through cluster analysis. *Appl Energy* 2010; 87: 442–51.
- [167] F. Bonanno, G. Capizzi, G. Graditi, C. Napoli, G. M. Tina. A radial basis function neural network based approach for the electrical characteristics estimation of a photovoltaic module. *Appl Energy* 2012; 97:956–61.
- [168] K. Ishaque, Z. Salam, S. Mekhilef, A. Shamsudin. Parameter extraction of solar photovoltaic modules using penalty-based differential evolution. *Appl Energy* 2012;99:297–308.
- [169] M. Ye, X. Wang, Y. Xu. Parameter extraction of solar cells using particle swarm optimization. *Journal of Applied Physics* 105 (9) (2009) 094502–094508.
- [170] N. Rajasekar, Neeraja Krishna Kumar, Rini Venugopalan. Bacterial Foraging Algorithm based solar PV parameter estimation. *Solar Energy*, Volume 97 (2013) 255–265.
- [171] E. W. Cowan. Basic Electromagnetism, *Academic Press*, New York, 1968.
- [172] H. R. Tizhoosh, Opposition-based reinforcement learning, *Journal of Advanced Computing Intelligence and Intelligent Informatics*, Volume 10, 3 (2006) 578–585.
- [173] M. Shokri, H. R. Tizhoosh and M. Kamel, *Opposition-based Q(k) algorithm*, *Proc. of IEEE World Congress on Computing Intelligence*, (2006) 646–653.
- [174] F. Wilcoxon, Individual comparisons by ranking methods, *Biometrics*, Volume 1 (1945) 80–83.
- [175] S. Garcia, D. Molina, M. Lozano and F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithm's behaviour: A case study on the CEC'2005 Special session on real parameter optimization, *Journal of Heuristics*, 2008.
- [176] J. Santamaría, O. Cerdón, S. Damas, J. M. García-Torres and A. Quirin, Performance evaluation of memetic approaches in 3D reconstruction of forensic objects, *Soft Computing*, 2008.
- [177] C. R. Johnson, M. G. Larimore. Comments on and Additions to an Adaptive Recursive LMS Filter. *Proceeding of the IEEE*, Volume 65 (1977) 1399–1402.
- [178] J. Bresenham. Linear algorithm for incremental digital display of circular arcs. *Communications of the ACM*, Volume 20 2 (1977) 100–106.
- [179] G. Boccignone, M. Ferraro, and P. Napoletano. Diffused expectation maximisation for image segmentation. *Electronics Letters*, Volume 40 18 (2004) 1107–1108.
- [180] G. Boccignone, P. Napoletano, V. Caggiano, and M. Ferraro. A multiresolution diffused expectation-maximization algorithm for medical image segmentation. *Computers in Biology and Medicine*, Volume 37 1 (2007) 83–96.
- [181] <http://www.cellavision.com/>
- [182] G. Landi and E. L. Piccolomini. An efficient method for nonnegatively constrained Total Variation-based denoising of medical images corrupted by Poisson noise. *Computerized Medical Imaging and Graphics*, Volume 36 1 (2012) 38–46.
- [183] W. Krattenthaler, K. J. Mayer, M. Zeiler. Point correlation: A reduced-cost template matching technique. In: *Proceedings of the First IEEE International Conference on Image Processing*, (1994) pp. 208–212
- [184] Z. Chunjiang, L. Xinyu, G. Liang, W. Qing. An improved electromagnetism-like mechanism algorithm for constrained optimization, *Expert Systems with Applications*, Volume 40 14 (2013) 5621–5634.
- [185] M. H. E. Pedersen. Good parameters for particle swarm optimization. Technical report HL1001. *Hvass Laboratories* (2010).
- [186] F. G. Lobo, C. F. Lima, and Z. Michalewicz, Eds. Parameter Setting in Evolutionary Algorithms. *Studies in Computational Intelligence*, Volume 54 (2007) Springer, Berlin, Germany.
- [187] C. B. B. Costa, M. R. W. MacIel, and R. M. Filho. Factorial design technique applied to genetic algorithm parameters in a batch cooling crystallization optimization. *Computers and Chemical Engineering*, Volume 29 10 (2005) 2229–2241.
- [188] A. Khadwilard, P. Luangpaiboon, and P. Pongcharoen. Full factorial experimental design for parameters selection of harmony search Algorithm. *The Journal of Industrial Technology*, Volume 8 2 (2012) 1–10.
- [189] G. E. P. Box, W. G. Hunter, and J. S. Hunter. Statistic for Experimenters: An Introduction to Design Data Analysis and Model Building, *John Wiley & Sons*, New York, NY, USA, 1978.
- [190] Lai and D. Tseng. A hybrid approach using Gaussian smoothing and genetic algorithm for

- multilevel thresholdingY, Volume 1 (2004) 143–152.
- [191] P.-Y. Yin. A fast scheme for optimal thresholding using genetic algorithms. *Signal Processing*, Volume 72 2 (1999) 85–95.
- [192] S. K. Pal, D. Bhandari, and M. K. Kundu. Genetic algorithms for optimal image enhancement. *Pattern Recognition Letters*, Volume 15 3 (1994) 261–2714.
- [193] R. L. Boylestad and Nashelsky L. Electrónica: teoría de circuitos y dispositivos electrónicos. *Pearson Education*, México, 2003.
- [194] Jun-hua Li, Ming Li. An analysis on convergence and convergence rate estimate of elitist genetic algorithms in noisy environments. *Optik*, Volume 124 (2013) 6780– 6785.
- [195] Hui Pan, Ling Wang, Bo Liu. Particle swarm optimization for function optimization in noisy environment. *Applied Mathematics and Computation*, Volume 181 (2006) 908–919.
- [196] H.G. Beyer. Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice. *Comput. Methods Appl. Mech. Engrg.* Volume 186 (2000) 239-267.